# KEC'S

## A Collection of 10 SETS

### Model Questions

**& T.U. Model Questions-Answers**

B.Sc.CSIT (IV Semester)

*All Subjects*

**with New Syllabus**

**Salient Features of this book**

- New Syllabus
- Model Questions-Answers
- Model Questions Sets

**KEC**

**Publication and Distribution Pvt. Ltd.**

Kathmandu, Nepal

Phone: 01-4168301

*Scanned by TapScanner*

## A Collection of 10 Sets Model Questions

B.Sc.CSIT (IV Semester)

Edition: 2076

*Publisher:* KEC Publication and Distributor Kathmandu Pvt. Ltd.
Phone: 01-4168301, 01-4241777
email: kecpubication14@gmail.com
Price: Rs. 295/-

## Contents

**Computer Networks (CSC258)** **1-43**

- New Syllabus ..... 1
- T.U. Model Questions-Answers SET I ..... 4
- Model Questions-Answers SET II ..... 21
- Model Questions Sets For Practice ..... 34

**Database Management System (CSC260)** **44-101**

- New Syllabus ..... 44
- T.U. Model Questions-Answers SET I ..... 46
- Model Questions-Answers SET II ..... 70
- Model Questions Sets For Practice ..... 87

**Theory of Computation (CSC257)** **102–141**

- New Syllabus ..... 102
- T.U. Model Questions-Answers SET I ..... 105
- Model Questions-Answers SET II ..... 118
- Model Questions Sets For Practice ..... 132

**Operating System (CSC259)** **142–189**

- New Syllabus ..... 142
- T.U. Model Questions-Answers SET I ..... 144
- Model Questions-Answers SET II ..... 162
- Model Questions Sets For Practice ..... 179

**Artificial Intelligence (CSC261)** **190–224**

- New Syllabus ..... 190
- T.U. Model Questions-Answers SET I ..... 193
- Model Questions Sets For Practice ..... 213

*Scanned by TapScanner*

# Computer Networks

**Syllabus**

Course Title: Computer Networks — Full Marks: 60 + 20 + 20

Course No: CSC258 — Pass Marks: 24 + 8 + 8

Nature of the Course: Theory + Lab — Credit Hrs: 3

Semester: IV

**Course Description:** This course introduces concept of computer networking and discuss the different layers of networking model.

**Course Objective:** The main objective of this course is to introduce the understanding of the concept of computer networking with its layers, topologies, protocols & standards, IPv4/IPv6 addressing, Routing and Latest Networking Standards

**Course Contents:**

**Unit 1: Introduction to Computer Network** **(6 Hrs.)**

1.1 Definitions, Uses, Benefits

1.2 Overview of Network Topologies (Star, Tree, Bus...)

1.3 Overview of Network Types (PAN, LAN, CAN, MAN...)

1.4 Networking Types (Client/Server, P2P)

1.5 Overview of Protocols and Standards

1.6 OSI Reference Model

1.7 TCP/IP Models and its comparison with OSI.

1.8 Connection and Connection-Oriented Network Services

1.9 Internet, ISPs, Backbone Network Overview

**Unit 2: Physical Layer and Network Media** **(4 Hrs.)**

2.1 Network Devices: Repeater, Hub, Switch, Bridge, Router

2.2 Different types of transmission Medias (wired: twisted pair, coaxial, fiber optic, Wireless: Radio waves, micro waves, infrared)

2.3 Ethernet Cable Standards (UTP & Fiber cable standards)

2.4 Circuit, Message & Packet Switching

2.5 ISDN: Interface and Standards

**Unit 3: Data Link Layer** **(8 Hrs.)**

3.1 Function of Data Link Layer (DLL)

3.2 Overview of Logical Link Control (LLC) and Media Access Control (MAC)

3.3 Framing and Flow Control Mechanisms

3.4 Error Detection and Correction techniques

3.5 Channel Allocation Techniques (ALOHA, Slotted ALOHA)

Scanned by TapScanner

3.6 Ethernet Standards (802.3 CSMA/CD, 802.4 Token Bus, 802.5 Token Ring)
3.7 Wireless LAN: Spread Spectrum, Bluetooth, Wi-Fi
3.8 Overview Virtual Circuit Switching, Frame Relay& ATM
3.9 DLL Protocol: HDLC, PPP **(10 Hrs.)**

**Unit 4: Network Layer**

4.1 Introduction and Functions
4.2 IPv4 Addressing & Sub-netting
4.3 Class-full and Classless Addressing
4.4 IPv6 Addressing and its Features
4.5 IPv4 and IPv6 Datagram Formats
4.6 Comparison of IPv4 and IPv6 Addressing
4.7 Example Addresses: Unicast, Multicast and Broadcast
4.8 Routing
  4.8.1 Introduction and Definition
  4.8:2 Types of Routing (Static vs. Dynamic, Unicast vs. Multicast, Link State vs. Distance Vector, Interior vs. Exterior)
  4.8.3 Path Computation Algorithms: Bellman Ford, Dijkstra's
  4.8.4 Routing Protocols: RIP, OSPF & BGP
4.9 Overview of IPv4 to IPv6 Transition Mechanisms
4.10 Overview of ICMP/ICMPv6&NATing
4.11 Overview of Network Traffic Analysis
4.12 Security Concepts: Firewall & Router Access Control **(6 Hrs.)**

**Unit 5: Transport Layer**

5.1 Introduction, Functions and Services
5.2 Transport Protocols: TCP, UDP and Their Comparisons
5.3 Connection Oriented and Connectionless Services
5.4 Congestion Control: Open Loop & Closed Loop, TCP Congestion Control
5.5 Traffic Shaping Algorithms: Leaky Bucket & Token Bucket
5.6 Queuing Techniques for Scheduling
5.7 Introduction to Ports and Sockets, Socket Programming **(7 Hrs.)**

**Unit 6: Application Layer**

6.1 Introduction and Functions
6.2 Web &HTTP
6.3 DNS and the Query Types
6.4 File Transfer and Email Protocols: FTP, SFTP, SMTP, IMAP, POP3



6.5 Overview of Application Server Concepts: Proxy, Web, Mail
6.6 Network Management: SNMP

**Unit 7: Multimedia &Future Networking** **(4 Hrs.)**

7.1 Overview Multimedia Streaming Protocols: SCTP
7.2 Overview of SDN and its Features, Data and Control Plane
7.3 Overview of NFV
7.4 Overview of NGN

**Laboratory Works:**

The lab activities under this subject should accommodate at least the following;

1. Understanding of Network equipment, wiring in details
2. OS (Ubuntu/CentOS/Windows) installation, practice on basic Networking commands (ifconfig/ipconfig, tcpdump, netstat, dnsip, hostname, route...)
3. Overview of IP Addressing and sub-netting, static IP setting on Linux/windows machine, testing
4. Introduction to Packet Tracer, creating of a LAN and connectivity test in the LAN, creation of VLAN and VLAN trunking.
5. Basic Router Configuration, Static Routing Implementation
6. Implementation of Dynamic/interior/exterior routing (RIP, OSPF, BGP)
7. Firewall Implementation, Router Access Control List (ACL)
8. Packet capture and header analysis by wire-shark (TCP, UDP, IP)
9. DNS, Web, FTP server configuration (shall use packet tracer, GNS3)
10. Case Study: Network Operation Center Visit (ISP, Telecom, University Network)
11. LAB Exam, Report and VIVA

**Text Books:**

1. Data Communications and Networking, 4th Edition, Behrouz A Forouzan. McGraw-Hill
2. Computer Networking; A Top Down Approach Featuring The Internet, 2nd Edition, Kurose James F., Ross W. Keith PEARSON EDUCATION ASIA

Scanned by TapScanner

## TRIBHUVAN UNIVERSITY

Institution of Science and Technology

Bachelor Level/Second Year/ Fourth Semester/ Science

**Computer Networks [CSC 258]**

Full Marks: 60
Pass Marks: 24
Time: 3 Hrs

***Candidates are required to give their answers in their own words as far as practicable.***

The figures in the margin indicate full marks.

## T.U. MODEL QUESTIONS-ANSWERS SET I

**Group 'A'**

**Long Answer Questions** **(2 ×10=20)**

**Attempt any two questions.**

1. **Suppose you are assigned to design a LAN for an office having 3 departments. Each department will have 50 computers locating in 10 rooms each equipped with 5 computers. Make your own justification while selecting connecting devices and accessories.**

**Solution:**

If I am assigned to design a LAN for an office having 3 departments; each department having 50 computers locating in 10 rooms each equipped with 5 computers. I will select following connecting devices and accessories to setup the required LAN.

1) **Router:** 1 manageable router to connect the LAN with WAN
2) **Switch:** 1 manageable (32 port = 3 vlan ×10 rooms + 1 inbound + 1 extra for future reference)
3) **Hub:** 3×10= 30 hubs each with 8 ports (8 ports = 1 inbound + 5 PC + 2 extra for future reference)
4) **Cable:** Cat 6 UTP cable

In order to connect with WAN I will prefer to have one manageable router since I am planning to use single subnet. Since router is manageable I can use it not only for routing but can be used for firewall, Access Control List, DNS, DHCP and so on. I will prefer to have one manageable switch having 32 ports because for the security of 3 different departments I will create 3 VLAN of 10 members for 10 rooms; in each room the backbone from switch will be extended by 8 port hub.



2. **Highlight on the importance of routing algorithm. Explain Distance Vector Routing algorithm and compare it with link state routing.**

**Solution:**

In order to transfer the packets from source to the destination, the network layer must determine the best route through which packets can be transmitted. Whether the network layer provides datagram service or virtual circuit service, the main job of the network layer is to provide the best route. The routing protocol provides this job. The routing protocol is a routing algorithm that provides the best path from the source to the destination. The best path is the path that has the "least-cost path" from source to the destination. Routing is the process of forwarding the packets from source to the destination but the best route to send the packets is determined by the routing algorithm. A routing algorithm is a set of step-by-step operations used to direct Internet traffic efficiently. When a packet of data leaves its source, there are many different paths it can take to its destination. The routing algorithm is used to determine mathematically the best path to take.

**Distance Vector Routing Algorithm**

It is a dynamic routing algorithm. A distance-vector routing protocol requires that a router inform its neighbors of topology changes periodically. The term distance vector refers to the fact that the protocol manipulates vectors (arrays) of distances to other nodes in the network. It makes use of Bellman Ford Algorithm for making routing tables. Problems of this algorithm are:

- Count to infinity problem which can be solved by splitting horizon.
- Good news spread fast and bad news spread slowly.
- Persistent looping problem i.e. loop will be there forever.

**Distance Vector Routing Algorithm Working:**

**Step 1) Initialization**

- At the beginning, each node knows the cost of itself and its immediate neighbor. [Node directly connected to it.]
- The distance of any entry that is not a neighbor is marked as infinite (unreachable).

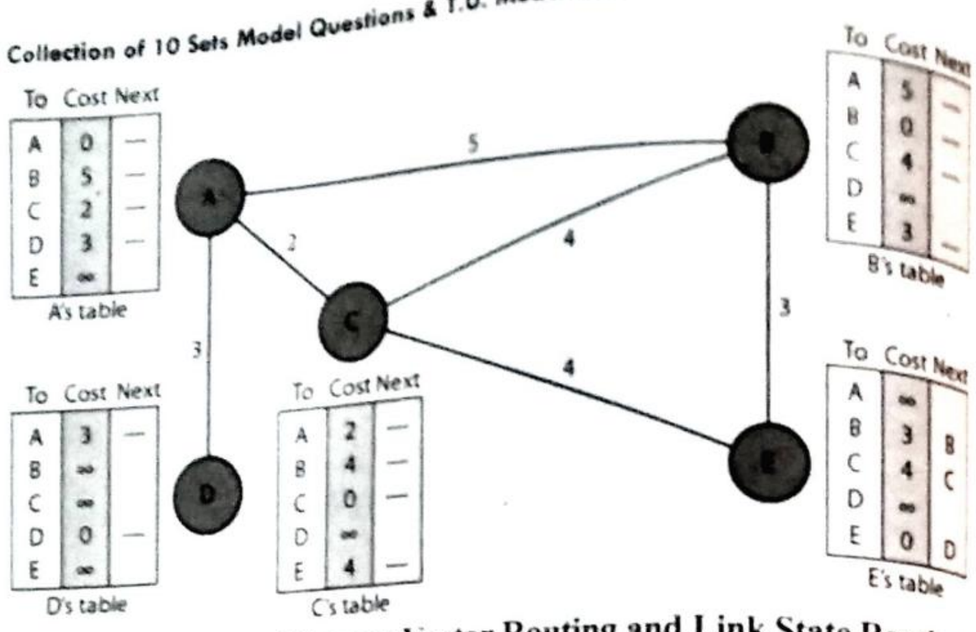
**Step 2) Sharing**

- Idea is to share the information between neighbors.
- The node C does not know the distance about D, but C's neighbor node A and node B does.
- If node A and B share it routing table with C, node C can also know how to reach node D.

**Step 3) Updating in distance vector routing**

- When node C get routing tables from A and B, it will compare the distance of both nodes to go to D, Whose ever distance is less, node C will update the entry in routing table according to it.

Scanned by TapScanner



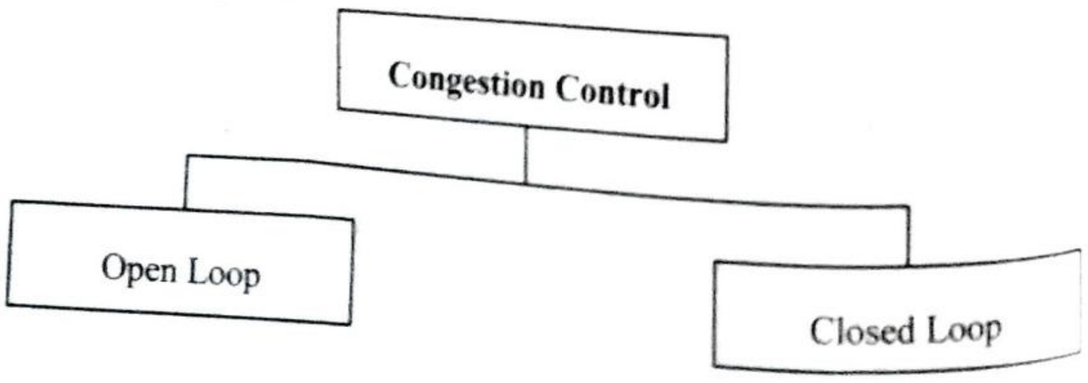

**Comparison between Distance Vector Routing and Link State Routing**

| Distance Vector Routing | Link State Routing |
|---|---|
| Bandwidth required is less due to local sharing, small packets and no flooding. | Bandwidth required is more due to flooding and sending of large link state packets. |
| Based on local knowledge since it updates table based on information from neighbors. | Based on global knowledge i.e. it have knowledge about entire network |
| Makes use of Bellman Ford algorithm | Make use of Dijkastra's algorithm |
| Traffic is less | Traffic is more |
| Converges slowly i.e. good news spread fast and bad news spread slowly. | Converges faster |
| Count to infinity problem | No count to infinity problem |
| Persistent looping problem i.e. loop will there forever. | No persistent loops, only transient loops. |
| Practical implementation is RIP and IGRP | Practical implementation is OSPF and ISIS |

**3. Explain various congestion control approaches.**

**Solution:**

Congestion control refers to the techniques used to control or prevent congestion. Congestion control techniques can be broadly classified into two categories:






**Open Loop Congestion Control**

Open loop congestion control policies are applied to prevent congestion before it happens. The congestion control is handled either by the source or the destination.

**Policies adopted by open loop congestion control –**

1) **Retransmission Policy:** It is the policy in which retransmission of the packets are taken care. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. This transmission may increase the congestion in the network. To prevent congestion, retransmission timers must be designed to prevent congestion and also able to optimize efficiency.
2) **Window Policy:** The type of window at the sender side may also affect the congestion. Several packets in the Go-back-n window are resent, although some packets may be received successfully at the receiver side. This duplication may increase the congestion in the network and making it worse. Therefore, Selective repeat window should be adopted as it sends the specific packet that may have been lost.
3) **Discarding Policy:** A good discarding policy adopted by the routers is that the routers may prevent congestion and at the same time partially discards the corrupted or less sensitive package and also able to maintain the quality of a message. In case of audio file transmission, routers can discard less sensitive packets to prevent congestion and also maintain the quality of the audio file.
4) **Acknowledgment Policy:** Since acknowledgement is also the part of the load in network, the acknowledgment policy imposed by the receiver may also affect congestion. Several approaches can be used to prevent congestion related to acknowledgment. The receiver should send acknowledgement for N packets rather than sending acknowledgement for a single packet. The receiver should send a acknowledgment only if it has to sent a packet or a timer expires.
5) **Admission Policy:** In admission policy a mechanism should be used to prevent congestion. Switches in a flow should first check the resource requirement of a network flow before transmitting it further. If there is a chance of congestion or there is congestion in the network, router should deny establishing a virtual network connection to prevent further congestion.

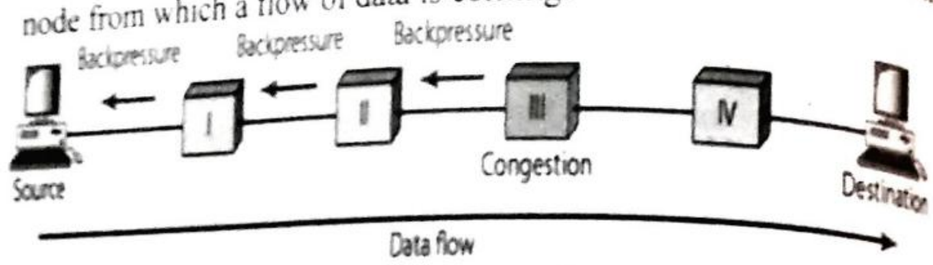
All the above policies are adopted to prevent congestion before it happens in the network.

**Closed Loop Congestion Control**

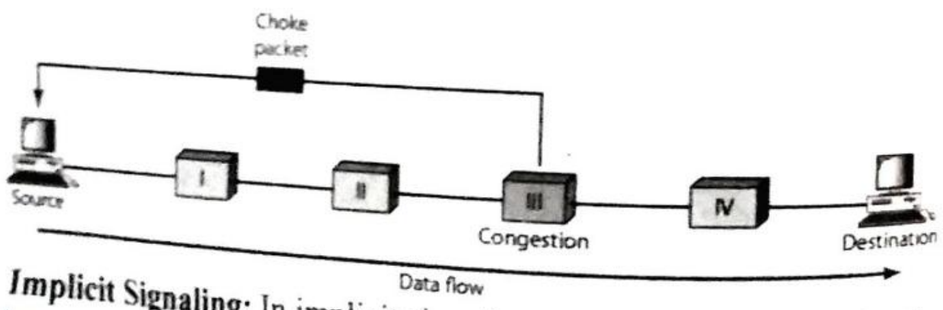
Closed loop congestion control technique is used to treat or alleviate congestion after it happens. Several techniques are used by different protocols; some of them are:

Scanned by TapScanner

1) **Back-pressure:** The technique of backpressure refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node or nodes. This may cause the upstream node or nodes to become congested, and they, in turn, reject data from their upstream nodes or nodes. And so on. Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source. The backpressure technique can be applied only to virtual circuit networks, in which each node knows the upstream node from which a flow of data is corning.




2) **Choke Packet:** A choke packet is a packet sent by a node to the source to inform it of congestion. Note the difference between the backpressure and choke packet methods. In backpressure, the warning is from one node to its upstream node, although the warning may eventually reach the source station. In the choke packet method, the warning is from the router, which has encountered congestion, to the source station directly. The intermediate nodes through which the packet has traveled are not warned. We have seen an example of this type of control in ICMP. When a router in the Internet is overwhelmed datagrams, it may discard some of them; but it informs the source. Host, using a source quench ICMP message. The warning message goes directly to the source station; the intermediate routers, and does not take any action.




3) **Implicit Signaling:** In implicit signaling, there is no communication between the congested node or nodes and the source. The source guesses that there is congestion somewhere in the network from other symptoms. For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested. The delay in receiving an acknowledgment is interpreted as congestion in the network; the source should slow down.



4) **Explicit Signaling:** The node that experiences congestion can explicitly send a signal to the source or destination. The explicit signaling method, however, is different from the choke packet method. In the choke packet method, a separate packet is used for this purpose; in the explicit signaling method, the signal is included in the packets that carry data. Explicit signaling, as we will see in Frame Relay congestion control, can occur in either the forward or the backward direction.

- **Backward Signaling**: A bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source that there is congestion and that it needs to slow down to avoid the discarding of packets.
- **Forward Signaling**: A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. The receiver in this case can use policies, such as slowing down the acknowledgments, to alleviate the congestion.

## Group B

**Short Answer Question Section**

**Attempt any EIGHT questions.** **(8×5=40)**

**4. Is 192.16.144.64/27 a host, network or broadcast address? In which layer of OSI model do HUB, Switch and Router operate on?**

**Solution:**

IP address: 192.16.144.64/27

Subnet Mask: 11111111 11111111 11111111 11100000
= 255.255.255.224

Total Subnets = $2^3 = 8$

Valid Subnets ($4^{th}$ octet) = Total subnet mask - current subnet mask
= 256 – 224= 32

Networks ID are: 192.16.144.0, 192.16.144.32, 192.16.144.64, 192.16.144.96, 192.16.144.128, 192.16.144.160, 192.16.144.192, 192.16.144.224

Therefore, 192.16.144.64/27 is a network address.

The layers of OSI model where HUB, Switch and Router operate are First Layer (Physical Layer), Second Layer (Data Link Layer) and Third Layer (Network Layer) respectively.

**5. Describe the working procedure of Token bus and Token ring.**

**Solution:**

**Token Bus**

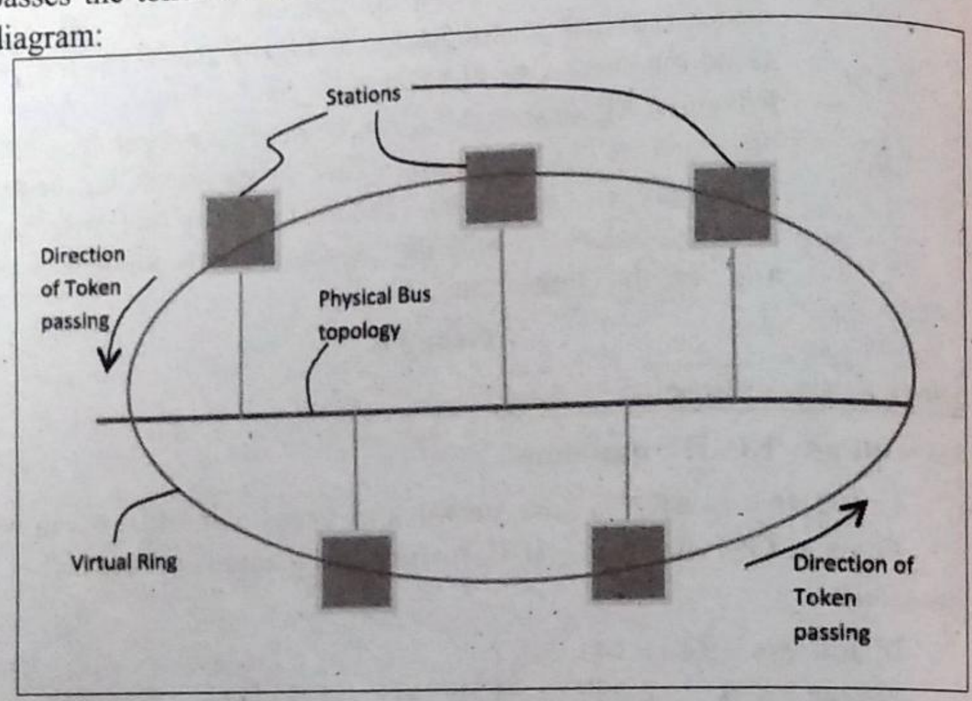
Token Bus (IEEE 802.4) is a standard for implementing token ring over virtual ring in LANs. The physical media has a bus or a tree topology and uses coaxial cables. A virtual ring is created with the nodes/stations and the token is passed from one node to the next in a sequence along

Scanned by TapScanner

this virtual ring. Each node knows the address of its preceding station and its succeeding station. A station can only transmit data when it has the token. The working principle of token bus is similar to Token Ring.

**Token Passing Mechanism in Token Bus**

A token is a small message that circulates among the stations of a computer network providing permission to the stations for transmission. If a station has data to transmit when it receives a token, it sends the data and then passes the token to the next station; otherwise, it simply passes the token to the next station. This is depicted in the following diagram:



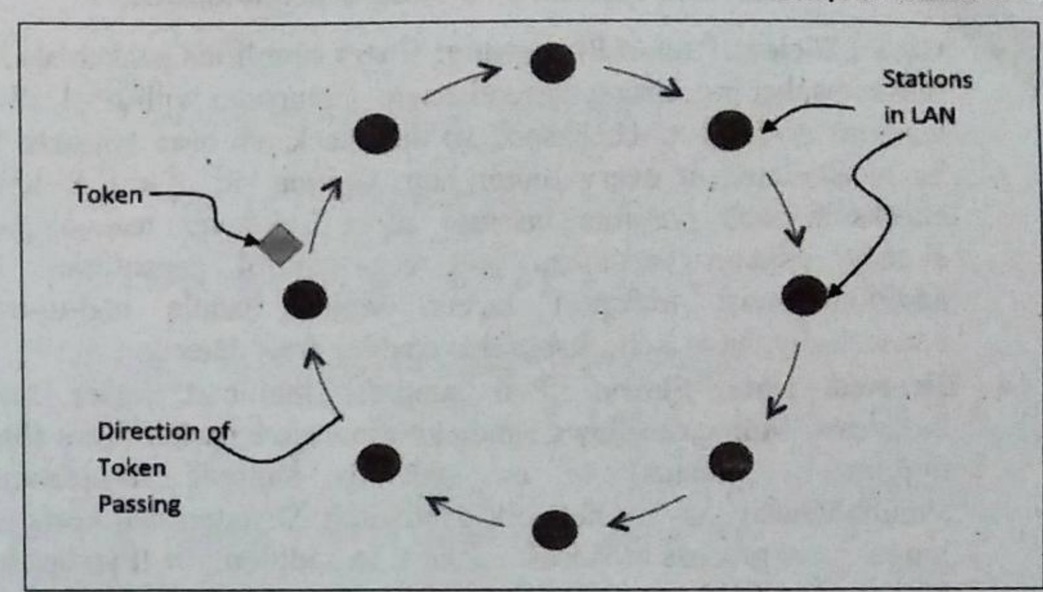

**Token Ring**

Token ring (IEEE 802.5) is a communication protocol in a local area network (LAN) where all stations are connected in a ring topology and pass one or more tokens for channel acquisition. A token is a special frame of 3 bytes that circulates along the ring of stations. A station can send data frames only if it holds a token. The tokens are released on successful receipt of the data frame.

**Token Passing Mechanism in Token Ring**

If a station has a frame to transmit when it receives a token, it sends the frame and then passes the token to the next station; otherwise it simply passes the token to the next station. Passing the token means receiving the token from the preceding station and transmitting to the successor station. The data flow is unidirectional in the direction of the token passing. In order that tokens are not circulated infinitely, they are removed from the network once their purpose is completed. This is shown in the following diagram:






**6. Why do you think network traffic analysis is carried out? How does IPv6 overcome the disadvantages of IPv4?**

**Solution:**

Unlimited information exchange is one of the most significant results of today's advancing computing and information technologies. The world becomes smaller, as people get more connected every day. This evolution has brought many benefits to our society when it comes to information dissemination, international cooperation, business opportunities and more. Unfortunately, these advantages bring with them serious security threats.

Information can be disseminated through unsecure avenues because anyone with basic knowledge of computers and internet computing can easily share information online. The various technological innovations have also given birth to a new generation of hackers, whose main objective is to steal and trade valuable information either for money or political purposes. This is why many companies have started to secure their firewalls, update their anti-malware software, and invest in a network security solutions. Network traffic analysis helps in warding off different cyber-attacks like:

- Uses Behavioral Patterns to Spot Suspicious Activities
- Helps Pinpoint Invisible Threats
- Minimizes Damage and Profit-loss

**IPv6 overcomes the disadvantages of IPv4 as follows:**

- **More Efficient Routing:** IPv6 reduces the size of routing tables and makes routing more efficient and hierarchical. IPv6 allows ISPs to aggregate the prefixes of their customers' networks into a single prefix and announce this one prefix to the IPv6 Internet. In addition, in IPv6 networks, fragmentation is handled by the source device, rather than the router, using a protocol for discovery of the path's maximum transmission unit (MTU).

Scanned by TapScanner

- **More Efficient Packet Processing:** IPv6's simplified packet header makes packet processing more efficient. Compared with IPv4, IPv6 contains no IP-level checksum, so the checksum does not need to be recalculated at every router hop. Getting rid of the IP-level checksum was possible because most link-layer technologies already contain checksum and error-control capabilities. In addition, most transport layers, which handle end-to-end connectivity, have a checksum that enables error detection.
- **Directed Data Flows:** IPv6 supports multicast rather than broadcast. Multicast allows bandwidth-intensive packet flows (like multimedia streams) to be sent to multiple destinations simultaneously, saving network bandwidth. Disinterested hosts no longer must process broadcast packets. In addition, the IPv6 header has a new field, named Flow Label that can identify packets belonging to the same flow.
- **Simplified Network Configuration**: Address auto-configuration (address assignment) is built in to IPv6. A router will send the prefix of the local link in its router advertisements. A host can generate its own IP address by appending its link-layer (MAC) address, converted into Extended Universal Identifier (EUI) 64-bit format, to the 64 bits of the local link prefix.
- **Support for New Services:** By eliminating Network Address Translation (NAT), true end-to-end connectivity at the IP layer is restored, enabling new and valuable services. Peer-to-peer networks are easier to create and maintain, and services such as VoIP and Quality of Service (QoS) become more robust.
- **Security:** IPSec, which provides confidentiality, authentication and data integrity, is baked into in IPv6. Because of their potential to carry malware, IPv4 ICMP packets are often blocked by corporate firewalls, but,ICMPv6, the implementation of the Internet Control Message Protocol for IPv6, may be permitted because IPSec can be applied to the ICMPv6 packets.

**7. Find Hamming Code for data 01100111.**

**Solution:**

Total number of data bits '$m$' = 8

Number of redundant bits $r$: $2^r >= m + r + 1$

$2^r \geq 8 + r + 1$

$2^4 \geq 8 + 4 + 1 = 13$ ***(Satisfied)***

Therefore, the value of $r$ is 4 that satisfy the above relation and $r_8 r_4 r_2 r_1$ are redundant bits.

Total number of bits = $m + r$ = 8 + 4 = 12;

| 0 | 1 | 1 | 0 | $r_8$ | 0 | 1 | 1 | $r_4$ | 1 | $r_2$ | $r_1$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

$r_1$: Checking even parity on bits 1,3,5,7,9,11



$r_1$: 1

$r_2$: Checking even parity on bits 2, 3, 6, 7, 10, 11

$r_2$:0

$r_4$: Checking even parity on bits 4, 5, 6, 7, 12

$r_4$:1

$r_8$: Checking even parity on bits 8, 9, 10, 11, 12

$r_8$:0

Substituting the value of redundant bits we get,

| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

Therefore, Hamming Code for data 01100111 is **011000111101.**

**8. Differentiate between frame relay and ATM.**

**Solution:**

The Differences between Frame Relay and ATM are as below:

1) The packet size in the frame relay varies while ATM uses a fixed size packet known as a cell.
2) ATM produces fewer overheads as compared to the frame relay technology.
3) Frame relay is less expensive respective to the ATM.
4) ATM is faster than the frame relay.
5) ATM provides error and flow control mechanism, whereas the frame relay does not provide it.
6) Frame relay is less reliable than the ATM.
7) Throughput generated by frame relay is medium. In contrast, ATM has a higher throughput.
8) The delay in the frame relay is more. As against, it is less in case of ATM.

**Comparison chart for Frame Relay Vs ATM**

| Basis for Comparison | Frame Relay | ATM |
|---|---|---|
| **Packet Size** | Variable | Fixed |
| **Processing Overhead** | Increased | Decreased |
| **Data Transfer** | Implemented in more than one area network. | Takes place within a LAN |
| **Cost** | Inexpensive | Cost is higher |
| **Speed** | Low | High |
| **QoS** | Quantifiable QoS is not provided. | Offers quantifiable QoS. |
| **Error Control** | No support is provided for error and flow control | Error and flow control is provided. |
| **Data Rate** | 64 Kbps up to 45 Mbps. | 155.5 Mbps or 622 Mbps. |
| **Reliability** | Low | Good |
| **Throughput** | Medium | High |
| **Delay** | High | Less |

**9. What is the function of proxy server? Explain about electronic mail.**

Scanned by TapScanner

**Solution:**

Proxy server is an intermediary server between client and the internet. Proxy server offers the following basic functionalities:

- Firewall and network data filtering.
- Network connection sharing
- Data caching

Electronic mail (also written as e-mail) is a digital message. Instead of using a pen to write a letter on paper, you're using your keyboard (or sometimes just your voice) to write an email message on an electronic device like a phone or computer.

Email addresses are written with a custom username at the beginning followed by the email service provider's domain name, with @ sign separating the two. For example: **name@mail.com**.

**Some other email basics:**

- **Sending the email:** When the email is finished and has been addressed to the recipient's email address, sending it allows the message to reach the intended recipient, or recipients if you're sending a group email.
- **Email transport:** Email servers transmit messages from the sender to the recipient. SMTP is the name of the protocol used to send email messages, and POP or IMAP servers are needed to understand how to download electronic mail to the email client.
- **Fetching new mail:** If we've got new mail in your mailbox, we simply have to tap it or click it to open and read the message, as well as to view any file attachments.

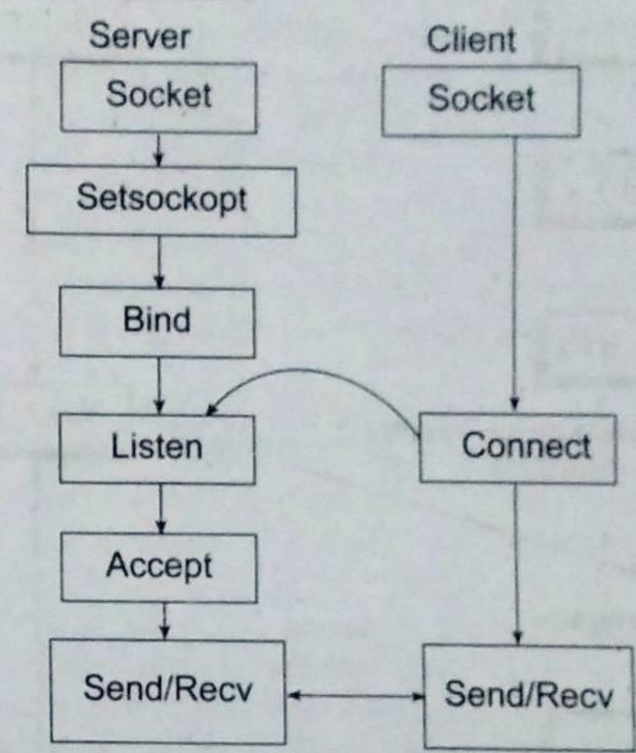
**10. Demonstrate the use of socket programming for creating network application using UDP and TCP with necessary diagrams.**

**Solution:**

If we are creating a connection between client and server using TCP then it has little functionality like, TCP is suited for applications that require high reliability, and transmission time is relatively less critical. It is used by other protocols like HTTP, HTTPs, FTP, SMTP, Telnet. TCP rearranges data packets in the order specified. There is absolute guarantee that the data transferred remains intact and arrives in the same order in which it was sent. TCP does Flow Control and requires three packets to set up a socket connection, before any user data can be sent. TCP handles reliability and congestion control. It also does error



checking and error recovery. Erroneous packets are retransmitted from the source to the destination.




The entire process can be broken down into following steps:

**TCP Server:**

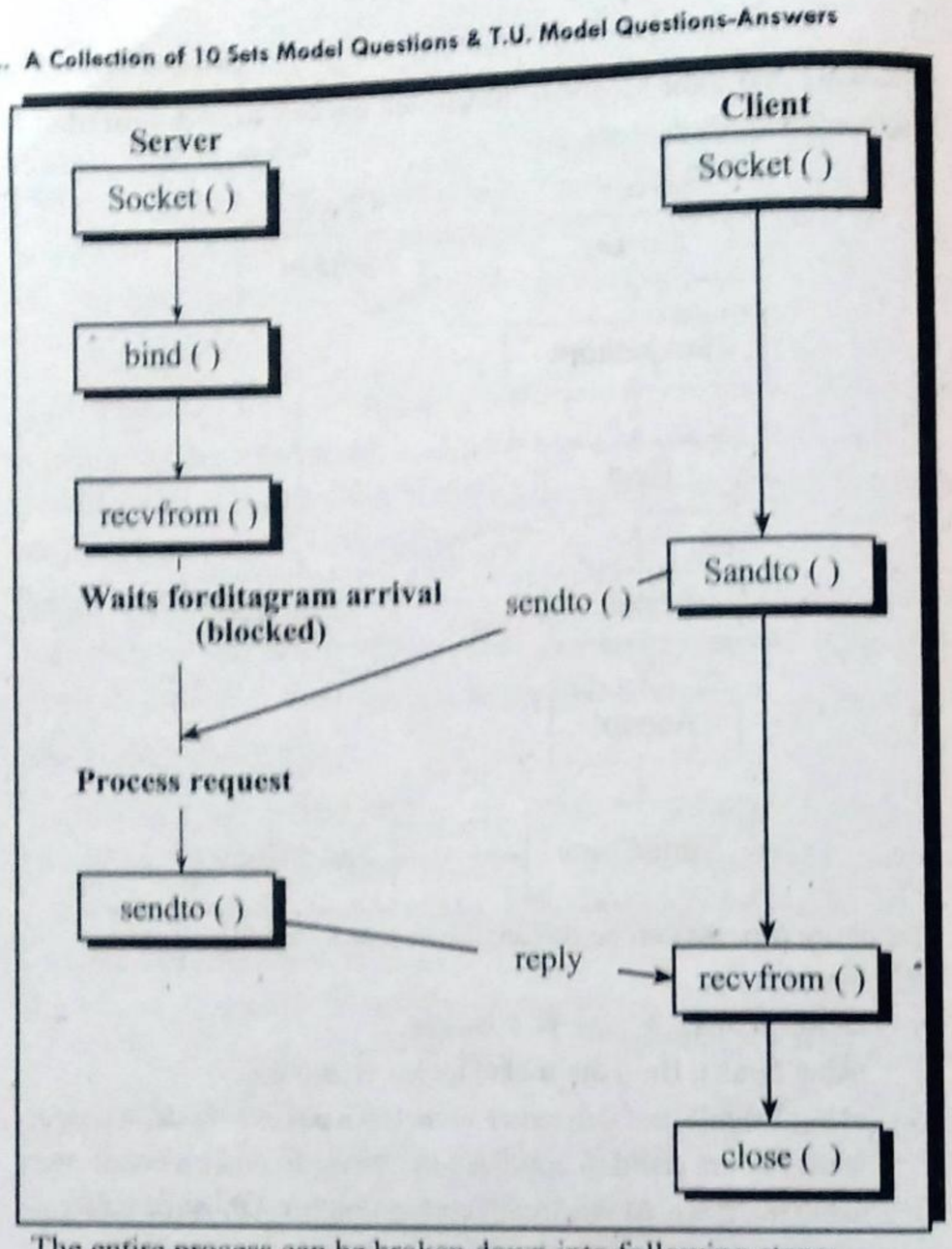
1) using create(), Create TCP socket.
2) using bind(), Bind the socket to server address.
3) using listen(), put the server socket in a passive mode, where it waits for the client to approach the server to make a connection
4) using accept(), At this point, connection is established between client and server, and they are ready to transfer data.
5) Go back to Step 3.

**TCP Client:**

1) Create TCP socket.
2) Connect newly created client socket to server.

In UDP, the client does not form a connection with the server like in TCP and instead just sends a datagram. Similarly, the server need not accept a connection and just waits for datagrams to arrive. Datagrams upon arrival contain the address of sender which the server uses to send data to the correct client.

Scanned by TapScanner




The entire process can be broken down into following steps:

**UDP Server:**

1) Create UDP socket.
2) Bind the socket to server address.
3) Wait until datagram packet arrives from client.
4) Process the datagram packet and send a reply to client.
5) Go back to Step 3.

**UDP Client:**

1) Create UDP socket.
2) Send message to server.
3) Wait until response from server is received.
4) Process reply and go back to step 2, if necessary.
5) Close socket descriptor and exit.



**11. Explain DNS with reference to its hierarchy and records.**

**Solution:**

The DNS stands for Domain Name System. The prominent intention of DNS is to translate domain names to the IP addresses. Even though there are there are domain names for all the websites, there are IP addresses also for them. The Internet uses this IP addresses to identify the websites. The IP address is a numerical data incorporated with four parts separated by dots (.). This numerical value is not easy to remember, so domain names were created, which are easily memorable. The DNS is responsible for translating these domain names to the IP addresses.

The DNS is a worldwide network that collectively forms a database of domain names and IP addresses. This database is a global one. The hierarchy consists of DNS servers. A DNS server can be defined as the following.

**DNS Server**

A DNS server is also a web server. Its primary objective is to interact with the aforementioned database. These DNS servers translate the domain name entered into the URL area of a web browser to the corresponding IP address. There are thousands of DNS servers worldwide which form the Domain Name System which currently is the largest digital database.

**Fully Qualified Domain Name (FQDN)**

It is essential to know about Fully Qualified Domain Name (FQDN), to understand the DNS hierarchy. A FQDN is the domain name that specifies its exact location in the DNS hierarchy. It specifies all domain levels including the top-level domain and the root zone. It consists of two parts, the host name and the domain name. An example of FQDN in a mail server is "mail.domain.com" where "mail" is the host name and the "domain.com" is the domain name. A fully qualified domain name is supposed to have little ambiguity. FQDN is otherwise called an absolute domain name.

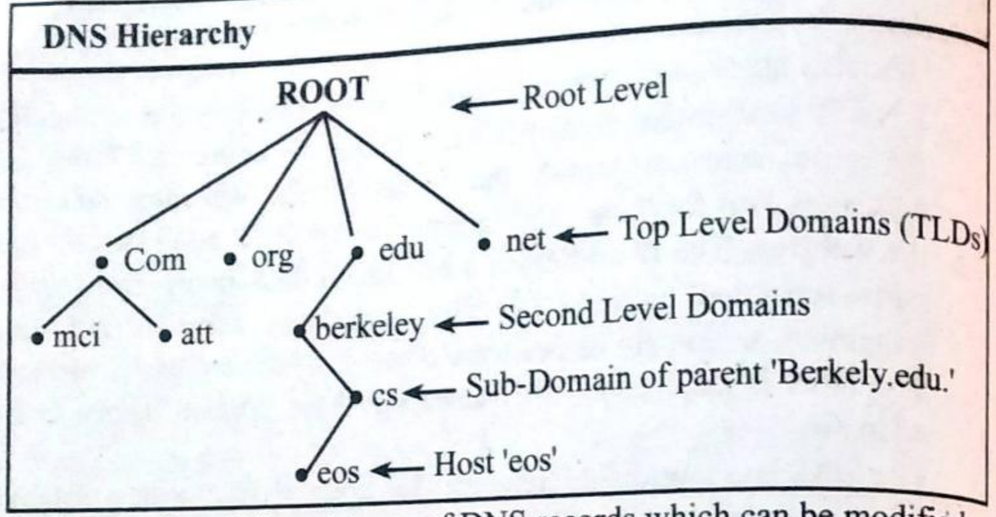
**DNS Hierarchy**

The DNS hierarchy is comprised of the following elements:

1) Root Level
2) Top Level Domains
3) Second Level Domains
4) Sub-Domain
5) Host

For example, in the figure below, .edu is the top level domain, berkeley is the second level domain, and .cs is the sub-domain of berkeley. Eos is the host name. A DNS server would store the IP address of the host where its name resides in the tree.

Scanned by TapScanner




There are a few different types of DNS records which can be modified at domain registrar. Typically, all we need to do is update the name server, but knowing the different types of records can help if we need to change something down the line.

**A Record**

An A record is what points your domain to an IP address. It stands for address record and is the purest form of DNS. It allows users to type in an easily recognizable domain and still gets pointed to the IP address.

**CNAME**

CNAME, or canonical name, redirects one domain to another, allowing you to only update one a record each time you make a change. For example, the CNAME record allows "cloudwards.net" to fetch up "www.cloudwards.net" with the "www" in front.

**MX Entry**

A mail exchanger entry directs email to a different server despite being a sub-domain. Essentially, it specifies how email should be routed when sent to an address at your domain.

**TXT Records**

This is a bit of a catch-all record, not intended to direct any traffic, but instead to provide information to external sources. It serves several different purposes depending on your needs.

**AAAA Record**

This record is the same as an A record, but allows you to point a domain to an IPv6 address instead of an IPv4 one.

Don't worry if all of this seems overwhelming. Routing data is complex, and these base descriptions barely scratch the surface. In most cases, you'll never need to touch these records and, if you do, most support can handle it for you.



**12. Write Short Notes (Any Two):**

**a. Firewall:** Firewall is a security barrier between two networks that screens traffic coming in and out of the gate of one network to accept or reject connections and services according to a set of rules. A firewall is like a secretary for a network which examines requests for access to the network. It decides whether they pass a reasonableness test. If they pass it they are allowed through and if not they are refused. If a man wants to meet the chair of the community department, the secretary does a certain level of filtering but if the man wants to meet the President of the country, the secretary will perform a much different level of filtering.

A network firewall is placed between the internal network, which might be considered safe, and the external network or the Internet which is known to be unsafe. The job of the firewall is to determine what to let into and out of the internal network. In this way, a firewall provides access control for the network. There are essentially three types of firewalls. Each type of firewall filters packets by examining the data up to a particular layer of the network protocol stack. The firewalls are:

- A packet filter is a firewall that operates at the network layer.
- A stateful packet filter is a firewall that lives at the transport layer.
- An application proxy is a firewall that operates at the application layer where it functions as a proxy.

**b. Packet Switching:** Packet switching is a method of transferring the data to a network in form of packets. In order to transfer the file fast and efficient manner over the network and minimize the transmission latency, the data is broken into small pieces of variable length, called Packet. At the destination, all these small-parts (packets) has to be reassembled, belonging to the same file. A packet composes of payload and various control information. No pre-setup or reservation of resources is needed.

Packet Switching uses Store and Forward technique while switching the packets; while forwarding the packet each hop first store that packet then forward. This technique is very beneficial because packets may get discarded at any hop due to some reason. More than one path is possible between a pair of source and destination. Each packet contains Source and destination address using which they independently travel through the network. In other words, packets belonging to the same file may or may not travel through the same path. If there is congestion at some path, packets are allowed to choose different path possible over existing network.

Packet-Switched networks were designed to overcome the weaknesses of Circuit-Switched networks since circuit-switched

Scanned by TapScanner

networks were not very effective for small messages. The packet switching has two approaches: **Virtual Circuit approach** and **Datagram approach.** WAN, ATM, frame relay and telephone networks use connection oriented virtual circuit approach; whereas internet relies on connectionless datagram based packet switching.

c. **NGN:** A next generation network is a packet-based network that can be used for both telephony and data and that supports mobility. Initially, the term Next Generation Network was used to refer to the transformation of the core network to IP. Sometimes a NGN is referred to as an all-IP network.

According to ITU "A Next Generation Network (NGN) is a packet-based network able to provide services including Telecommunication Services and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent from underlying transport-related technologies. It offers unrestricted access by users to different service providers. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users."

**The NGN is characterized by the following fundamental aspects:**

- Packet-based transfer
- Separation of control functions among bearer capabilities, call/session, and application/ service
- Decoupling of service provision from network, and provision of open interfaces
- Support for a wide range of services, applications and mechanisms based on service building blocks (including real time/ streaming/ non-real time services and multi-media)
- Broadband capabilities with end-to-end QoS and transparency
- Interworking with legacy networks via open interfaces
- Generalized mobility
- Unrestricted access by users to different service providers
- A variety of identification schemes which can be resolved to IP addresses for the purposes of routing in IP networks
- Unified service characteristics for the same service as perceived by the user
- Converged services between Fixed/Mobile
- Independence of service-related functions from underlying transport technologies



## MODEL QUESTIONS-ANSWERS SET II

Bachelor Level/Second Year/ Fourth Semester/ Science — Full Marks: 60
**Computer Networks [CSC 258]** — Pass Marks: 24
Time: 3 Hrs

***Candidates are required to give their answers in their own words as far as practicable.***

The figures in the margin indicate full marks.

**Group 'A'**

**Long Answer Questions**

**Attempt any two questions.** **(2 ×10=20)**

1. **Suppose a router has build up the routing table shown below. The router can deliver packets directly over interfaces 0 and 1, or it can forward packets to routers R2, R3 or R4. Describe what the router does with a packet addressed to each of the following destinations.**

   a. **128.96.39.10** b. **128.96.40.12**
   c. **128.96.40.151** d. **192.4.153.17**
   e. **192.4.153.90**

**Routing Table**

| Subnet Number | Subnet Mask | Next Hop |
|---|---|---|
| 128.96.39.0 | 255.255.255.128 | Interface 0 |
| 128.96.39.128 | 255.255.255.128 | Interface 1 |
| 128.96.40.0 | 255.255.255.128 | R2 |
| 192.4.153.0 | 255.255.255.192 | R3 |
| (Default) | | R4 |

**Solution:**

i. 128.96.39.10-Applying the subnet mask 255.255.255.128, we get 128.96.39.0. Use the interface 0 as the next hop.

ii. 128.96.40.12-Applying the subnet mask 255.255.255.128, we get 128.96.40.0 (next hop is Router R2)

iii. 128.96.40.151- Applying the entire subnet mask give 128.96.40.128, none of the subnet number entries match, hence use default Router R4.

iv. 192.4.153.17-Applying subnet mask 255.255.255.192, we get 192.4.153.0. Use Router R3 as next hop.

v. 192.4.153.90- None of the subnet number entries match hence uses default Router R4 next hop.

Scanned by TapScanner

**2. Why do we need layered protocol architecture? Explain the Functions of 7 layers of OSI model.**

**Solution:**

A layered architecture facilitates development in complex environments by grouping specific related functions into separate well-defined layers with clear interfaces. This methodology reduces complexity by breaking the problem space into smaller and simpler components and standardizes interfaces facilitating multi-vendor development and modular component-based engineering. Layered architectures in conjunction with open standards define a common vocabulary necessary for understanding and cooperation in multi-vendor environments and positively results in increased competition and innovation. The need of layered protocol architecture can be summarized as follow:

- To Separate Specific Functions in Each Layer.
- Each Layer Should Define a Unique Function.
- To make their Implementation Transparent to Other Components.
- Allows Independent Design and Testing of Each Component.
- Modularization Eases Maintenance and Updating of System.

**Functions of 7 layers of OSI model are as follows:**

**1) Physical Layer: 1**

- Physical characteristics of interfaces and media
- Representation of bits
- Data Rate or Transmission rate
- Synchronization of bits
- Line Configuration
- Physical Topology
- Transmission Mode

**2) Data Link Layer:**

- Enable Node to Node (hop to hop) communication
- Responsible for transmitting frames from one node to next.
- Framing
- Physical addressing
- Flow Control
- Error control
- Access Control [E.g. CSMA/CD]

**3) Network Layer:**

- Enables Host to Host Communication
- Responsible for Delivery of Packets
- Logical Addressing
- Routing

**4) Transport Layer:**

- Enables Process to Process Communication.
- Port Addressing
- Segmentation and Reassembly.



- Connection Control
- Flow Control
- Error Control

**5) Session Layer:**

- Control of Dialogues between Applications.
- Whose Turn is to Transmit?
- Dialogue Discipline => Half Duplex/ Full Duplex

**6) Presentation Layer:**

- Data Formats and Coding.
- Data Compression
- Encryption

**7) Application Layer:**

- Responsible for Providing Service to End Users.
- Mail Transfer Service.
- File Transfer Service.

**3. What is routing? Discuss link state routing algorithm in detail.**

**Solution:**

Routing is the path that network data or a packet takes to reach its destination on a network. The network router is what decides the best route for each network packet. There are a lot of different routing protocols, each one designed for specific network sizes. The two types of routing are **static routing** and **dynamic routing**.

Link state routing is the second family of routing protocols. While distance vector routers use a distributed algorithm to compute their routing tables, link-state routing uses link-state routers to exchange messages that allow each router to learn the entire network topology. Based on this learned topology, each router is then able to compute its routing table by using a shortest path computation.

**Features of link state routing protocols:**

- **Link state packet:** A small packet that contains routing information.
- **Link state database:** Collection information gathered from link state packet.
- **Shortest path first algorithm (Dijkstra algorithm):** A calculation performed on the database results into shortest path
- **Routing table:** A list of known paths and interfaces.

**Calculation of shortest path:**

To find shortest path, each node need to run the famous **Dijkstra algorithm**. This famous algorithm uses the following steps:

Scanned by TapScanner

- **Step-1:** The node is taken and chosen as a root node of the tree, this creates the tree with a single node, and now set the total cost of each node to some value based on the information in Link State Database
- **Step-2:** Now the node selects one node, among all the nodes not in the tree like structure, which is nearest to the root, and adds this to the tree. The shape of the tree gets changed.
- **Step-3:** After this node is added to the tree, the cost of all the nodes not in the tree needs to be updated because the paths may have been changed.
- **Step-4:** The node repeats the Step 2. and Step 3. until all the nodes are added in the tree

**Group 'B'**

**Long Answer question Section**

**Attempt any EIGHT questions.** [8×5=40]

**4. Explain ATM Architecture.**

**Solution:**

The asynchronous transfer mode (ATM) protocol architecture is designed to support the transfer of data with a range of guarantees for quality of service. The user data is divided into small, fixed-length packets, called cells, and transported over virtual connections. ATM operates over high data rate physical circuits, and the simple structure of ATM cells allows switching to be performed in hardware, which improves the speed and efficiency of ATM switches.

Figure below shows the reference model for ATM. The first thing to notice is that, as well as layers, the model has planes. The functions for transferring user data are located in the user plane; the functions associated with the control of connections are located in the control plane; and the co-ordination functions associated with the layers and planes are located in the management planes.

The three-dimensional representation of the ATM protocol architecture is intended to portray the relationship between the different types of protocol. The horizontal layers indicate the encapsulation of protocols through levels of abstraction as one layer is built on top of another, whereas the vertical planes indicate the functions that require co-ordination of the actions taken by different layers. An advantage of dividing the functions into control and user planes is that it introduces a degree of independence in the definition of the functions; the protocols for transferring user data (user plane) are separated from the protocols for controlling connections (control plane).



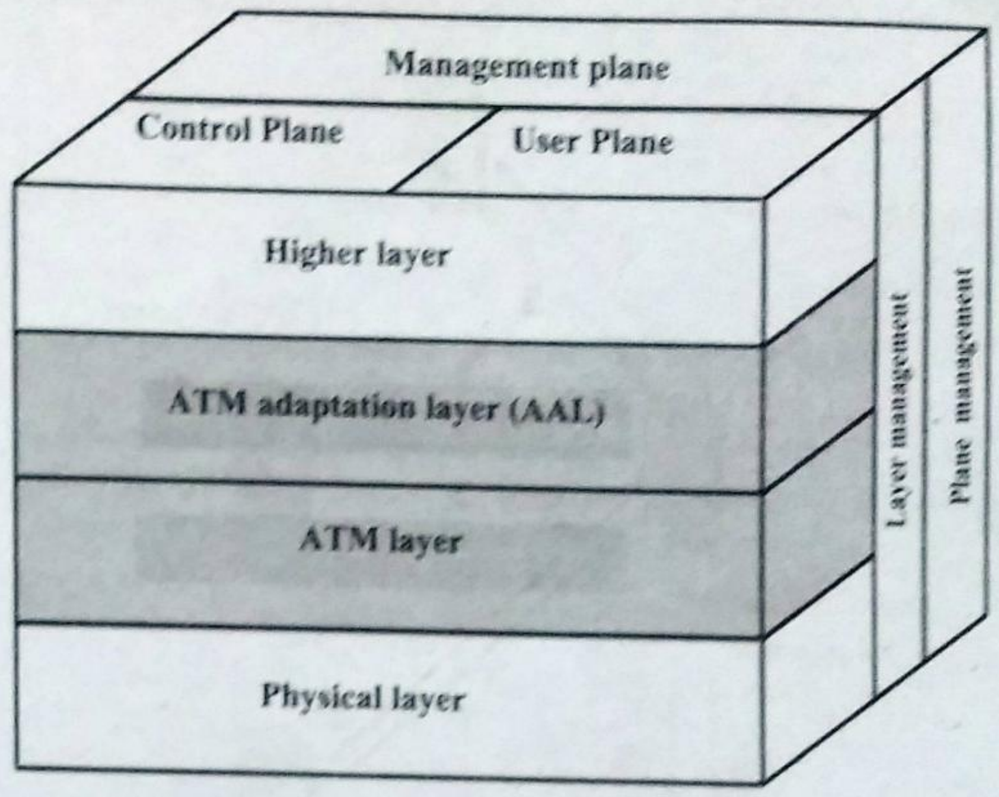



**Fig.:** ATM Reference Model

The protocols in the ATM layer provide communication between ATM switches while the protocols in the ATM adaptation layer (AAL) operate end-to-end between users.

**5. What is DNS? Explain its working with the example.**

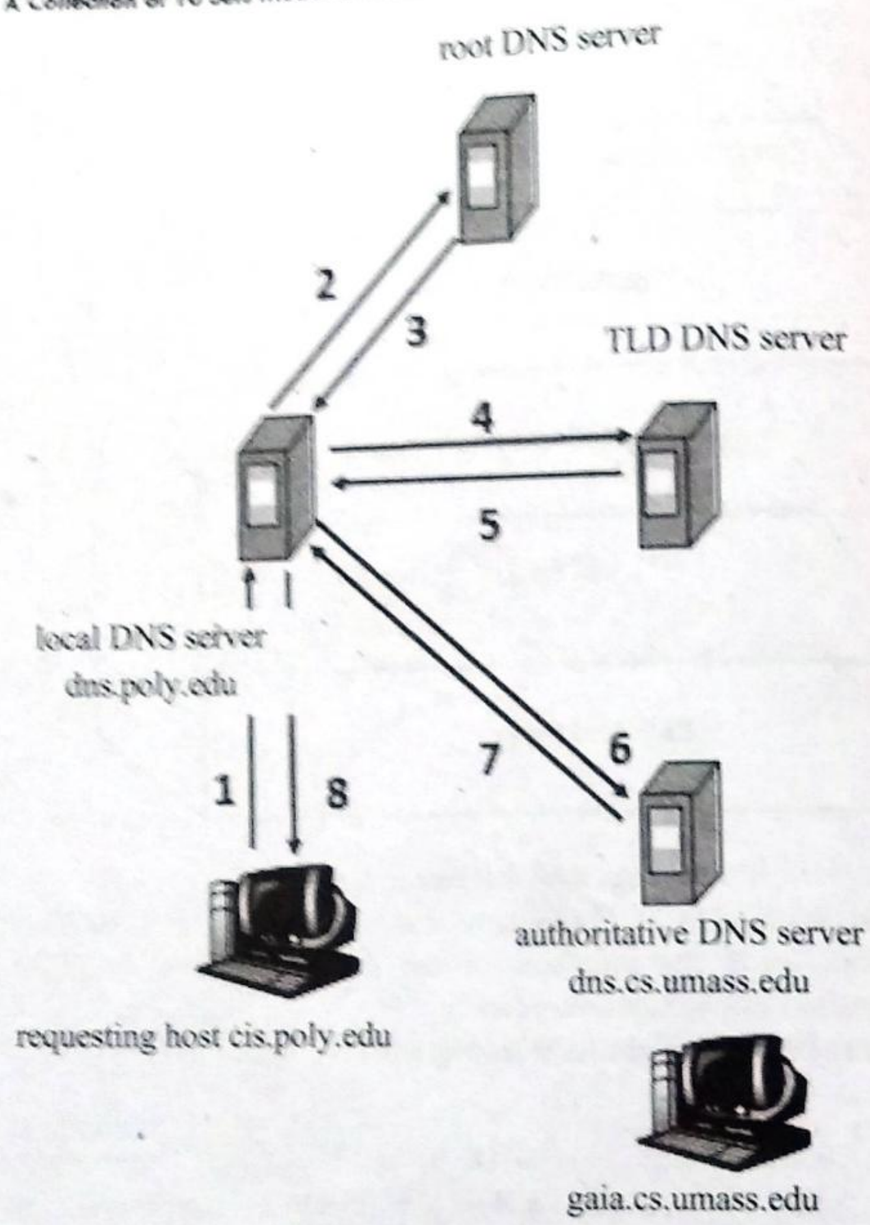
**Solution:**

**DNS:** Domain Name Servers (DNS) are the Internet's equivalent of a phone book. They maintain a directory of domain names and translate them to Internet Protocol (IP) addresses. This is necessary because, although domain names are easy for people to remember, computers or machines, access websites based on IP addresses.

**Example:**

The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer. If the server is the authority for the domain name, it checks its database and responds. If the server is not the authority, it sends the request to another server (the parent usually) and waits for the response. If the parent is the authority, it responds; otherwise, it sends the query to yet another server. When the query is finally resolved, the response travels back until it finally reaches the requesting client. This is called recursive resolution and is shown in Figure below.

Scanned by TapScanner




**6. Differentiate between TCP and UDP.**

**Solution:**

Difference between TCP and UDP are shown in table below:

| Characteristics / Description | UDP | TCP |
|---|---|---|
| General | Simple, High Speed, Low Functionality | Full-featured, Reliable Data Transfer |
| Connection Setup | Connectionless | Connection Oriented |
| Data Interface to Application | Message Based Data is Sent | Stream Based data is sent |
| Reliability | Unreliable (No ACK) | Reliability (ACK available) |



| | | |
|---|---|---|
| Retransmission | Not Performed | Data is Lost → Retransmission |
| Flow Control | None | Sliding Window → Flow Control |
| Overhead | Very Low | Low, Higher than UDP |
| Transmission Speed | Very High | High, Lower than UDP |
| Application that uses Protocol | DNS, DHCP, SNMP, (Multimedia Application) | FTP, Telnet, SMTP, DNS, HTTP, etc. |

**7. A company is granted a site address 201.70.64.0. The company needs six subnets. Design the subnets (Subnet masks for each subnet, starting and ending address of each subnet).**

**Solution:**

Given IP address: 201.70.64.0

Subnet Mask: 11111111 11111111 11111111 00000000 = 255.255.255.0

Required subnet = 6

Now, $2^n$ > = required subnet

$2^3$> = 6 (When n=3)

8> =6 (True)

Therefore, to create 6 subnet 3 host bits needed to be converted into network bit

New network bits = 24+3 =27

New IP address: 201.70.64.0 /27

New Subnet Mask: 11111111 11111111 11111111 11100000 = 255.255.255.224

Total Subnets = $2^3$ = 8

Total hosts = $2^5$ = 32

Usable hosts = 32 – 2 = 30

Valid Subnets ($4^{th}$ octet) = 256 – 224= 32

| Starting Address | Range | | Ending Address |
|---|---|---|---|
| | First host | Last host | |
| 201.70.64.0 | 201.70.64.1 | 201.70.64.30 | 201.70.64.31 |
| 201.70.64.32 | 201.70.64.33 | 201.70.64.62 | 201.70.64.63 |
| 201.70.64.64 | 201.70.64.65 | 201.70.64.94 | 201.70.64.95 |
| 201.70.64.96 | 201.70.64.97 | 201.70.64.58 | 201.70.64.159 |
| 201.70.64.160 | 201.70.64.161 | 201.70.64.190 | 201.70.64.191 |
| 201.70.64.192 | 201.70.64.193 | 201.70.64.222 | 201.70.64.223 |

**8. What is Socket? Which are various socket primitives used in client and server communication?**

**Solution:**

Scanned by TapScanner

**Socket:** A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to. An endpoint is a combination of an IP address and a port number.

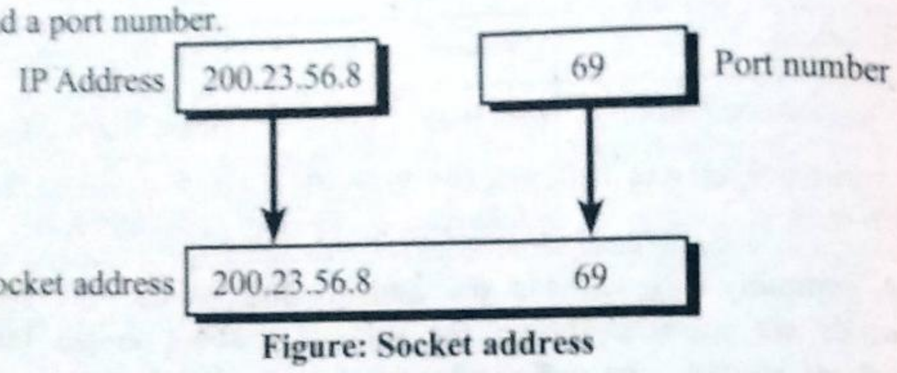



**Figure: Socket address**

**Socket Primitives:**

| Primitive | Meaning |
|---|---|
| SOCKET | Create a new connection end point |
| BIND | Attach a local address to a socket |
| LISTEN | Announce willingness to accept connections; give queue size |
| ACCEPT | Block the caller unit a connection attempt arrives |
| CONNECT | Actively attempt to establish a connection |
| SEND | Send some data over the connection |
| RECEIVE | Receive some data from the connection |
| CLOSE | Release the connection |

**9. Differentiate between IPv4 and IPv6 with header formats.**

**Solution:**

The Different between IPv4 and IPv6 with header formats are follows:

| Feature | IPv4 | IPv6 |
|---|---|---|
| Source and destination address | 32 bits | 128 bits |
| Address Format | Dotted Decimal | Hexadecimal Notation |
| No of Address | $2^{32}$ | $2^{128}$ |
| IPSec | Optional | required |
| Payload ID for QoS in the header | No identification | Using Flow label field |
| Fragmentation | Both router and the sending hosts | Only supported at the sending hosts |
| Header checksum | included | Not included |
| Resolve IP address to a link layer address | Broadcast ARP request | Multicast Neighbor solicitation message |



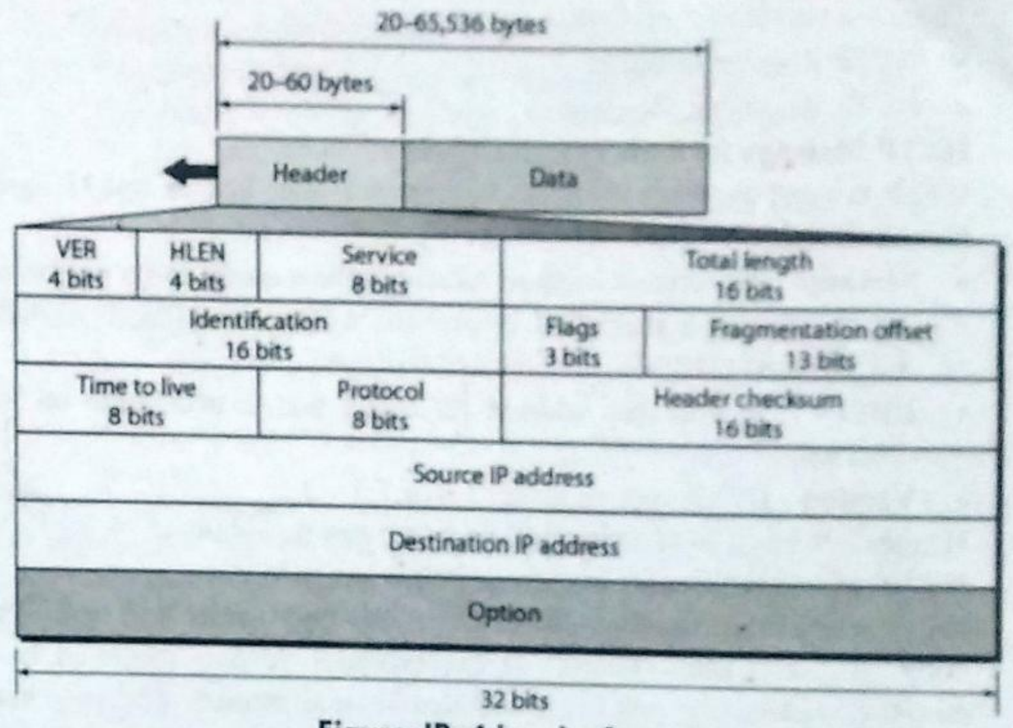



**Figure: IPv4 header format**

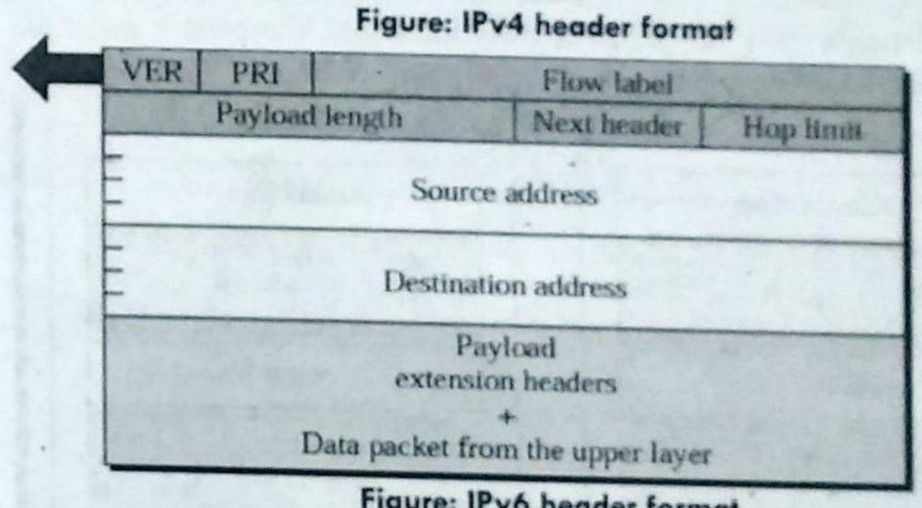



**Figure: IPv6 header format**

**10. What is difference between persistent and non persistent HTTP? Also Explain HTTP message format.**

**Solution:**

Difference between persistent and non persistent HTTP is also follows:

| Non Persistent HTTP | Persistent HTTP |
|---|---|
| Server closes connection after sending response | Server leaves connection open after sending response |
| Requires 2 RTTs per object | One RTT for all the referenced objects |
| New connection must be established and maintained for each requested object | client sends requests as soon as it encounters a referenced object |
| OS overhead for *each* TCP connection | No overhead for *each* TCP connection |
| Slow response | Faster in response |
| At most one object is sent over a TCP connection. | Multiple objects can be sent over single TCP connection between client and server |

Scanned by TapScanner

There are two HTTP messages

- HTTP Request message
- HTTP Response Message

**HTTP Message formats are given below**

HTTP request message contains request line and header line. Request line has 3 fields

- **Method-** The request method indicates the method to be performed on the resource identified by the given Request-URI. It includes GET, POST, HEAD
- **URL** - It is a unique address for a file that is accessible on that internet.
- **Version** - HTTP version either 1.0 or 1.1

Header line has 2 fields, header field name and its value.

**HTTP Message formats are given below**

HTTP response message contains status line and header line and entity body. A Status-Line consists of the protocol version followed by a numeric status code and its associated textual phrase. The response-header fields allow the server to pass additional information about the response which cannot be placed in the Status- Line.

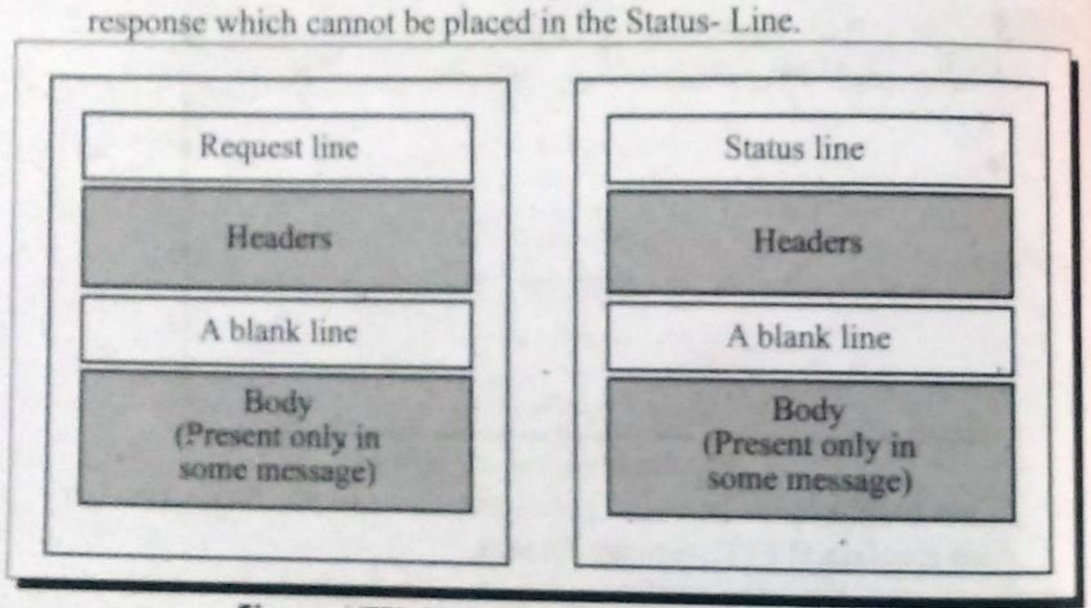



**Figure:** HTTP Request and Response message

**11. Explain Software-defined networking (SDN).**

**Solution:**

Software-defined networking (SDN) is an approach to computer networking that allows network administrators to manage network services through abstraction of higher-level functionality. This is done by decoupling the system that makes decisions about where traffic is sent (the control plane) from the underlying systems that forward traffic to the selected destination (the data plane). SDN requires some method for the control plane to communicate with the data plane. One such mechanism is OpenFlow.



- Separate Control plane and Data plane entities
  - Network intelligence and state are logically centralized
  - The underlying network infrastructure is abstracted from the applications
- Execute or run Control plane software on general purpose hardware
  - Decouple from specific networking hardware
  - Use commodity servers
- Have programmable data planes
  - Maintain, control and program data plane state from a central entity
- An architecture to control not just a networking device but an entire network

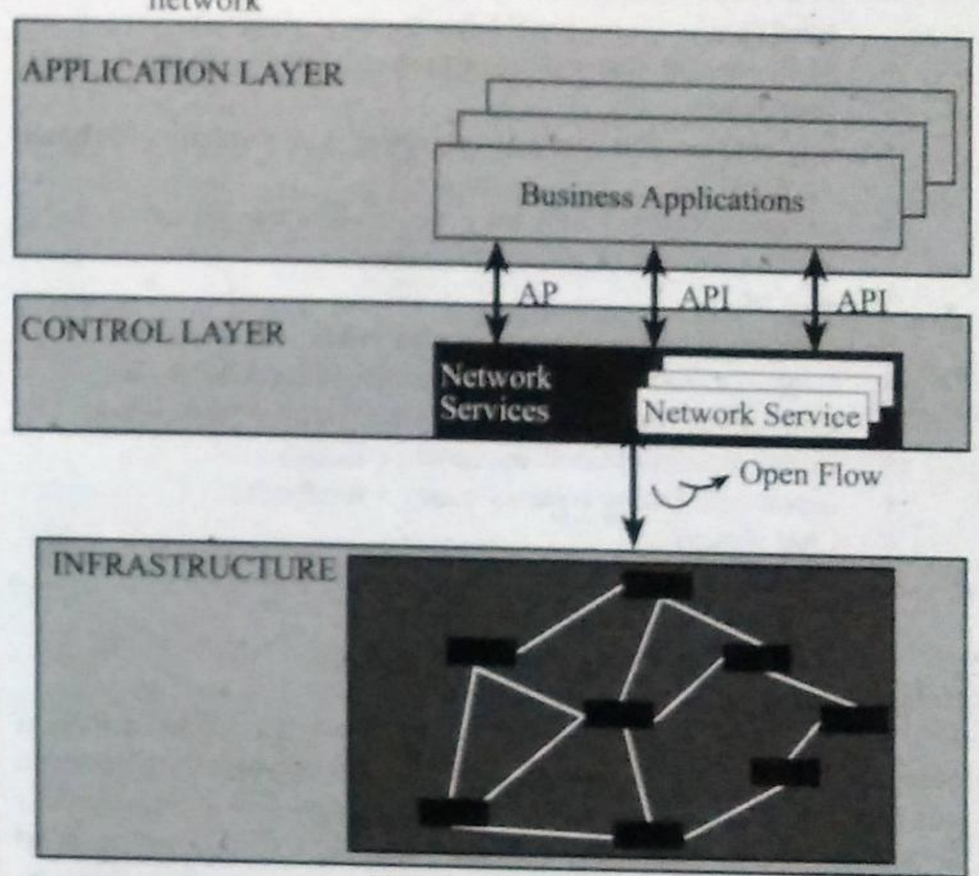



Figure: SDN Architecture

**SDN Benefits:**

- **Dynamic, Manageable, cost-effective, adaptable**
- **Directly programmable**
- **Agile:** administrators dynamically adjust network-wide traffic flow to meet changing needs.
- **Centrally managed**
- **Programmatically configured:** SDN lets network managers configure, manage, secure, and optimize network resources very quickly via dynamic, automated SDN programs
- **Open standards-based and vendor-neutral**

Scanned by TapScanner

12. **Write short notes on (Any two):**

a. **Multicast routing:** In unicast routing there is a single sender (source) and a single receiver (destination). In unicast routing, the router forwards the received packet through only one of its interfaces. In multicasting there is at least one sender and several receivers (group of receivers called multicast group). In multicast routing, the router may forward the received packet through several of its interfaces. Emulation of multicasting through multiple unicast routing is not efficient and may create long delays, particularly with a large group.

**Application of multicast routing:**

- One to Many
  - Scheduled audio-video distribution (lectures, business TV, meetings)
  - Push media (news headlines, weather updates, radio broadcast)
  - File distribution and caching (web site content, file-based updates)
  - Announcements (network time, session schedules)
  - Monitoring (stock prices, telemetry)
- Many to Many
  - Multimedia conferencing (audio/video, whiteboards)
  - Synchronized resources (shared distributed databases)
  - Concurrent processing (distributed parallel processing)
  - Collaboration (shared document editing)
  - Distance learning (one-to-many + feedback)
  - Chat groups
  - Distributed interactive simulations
  - Multiplayer games
  - Jam sessions (shared encoded music)

b. **Point-to-point protocol**

PPP (Point-to-Point Protocol) is a protocol for communication between two computers using a serial interface, typically a personal computer connected by phone line to a server. For example, your Internet server provider may provide you with a PPP connection so that the provider's server can respond to your requests, pass them on to the Internet, and forward requested Internet responses back to you. PPP uses the Internet protocol (IP) (and is designed to handle others). It is sometimes considered a member of the TCP/IP suite of protocols. Relative to the Open Systems Interconnection (OSI) reference model, PPP provides layer (data-link layer) services. Essentially, it packages your computer's TCP/IP packets and forwards them to the server where they can actually be put on the Internet. PPP is a full-duplex protocol that can be used on various physical media, including twisted pair or fiber optic lines or satellite transmission. It uses a variation of High Speed Data Link Control (HDLC) for packet encapsulation. PPP can handle synchronous as



well as asynchronous communication. PPP can share a line with other users and it has error detection that SLIP lacks. Where a choice is possible, PPP is preferred.

c. **ICMP**

Since IP does not have an inbuilt mechanism for sending error and control messages. It depends on Internet Control Message Protocol (ICMP) to provide an error control. ICMP is a protocol used primarily for sending error messages, performing diagnostics, and controlling the flow of data. An example of an error message and of flow control is an ICMP source-quench packet sent by a router to a source host to tell the host to slow down because the router is overloaded. Performing Diagnostics with ICMP: As stated earlier, the ICMP protocol is used for performing diagnostics. An example of using ICMP as a diagnostic tool is with the Ping utility. Ping stands for Packet Internet Groper.

**ICMP messages are divided into two broad categories:**

- Error-reporting messages
- Query messages.

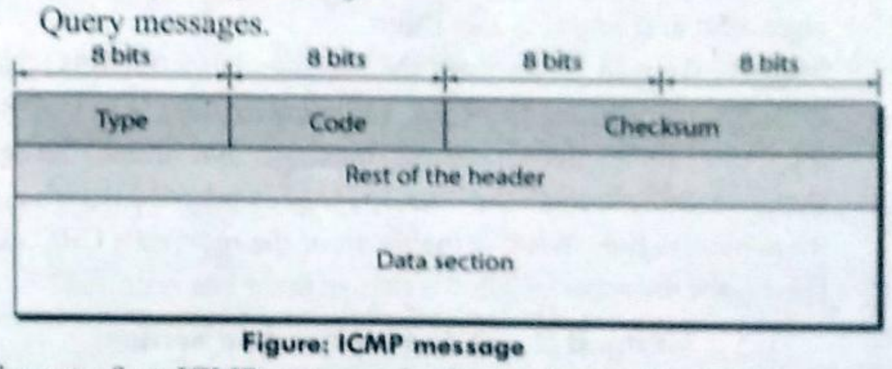



**Figure: ICMP message**

The format of an ICMP message is shown above. The 8-bit type code identifies the types of message. Each ICMP message contains three fields that define its purpose and provide a checksum. They are TYPE, CODE, and CHECKSUM fields. The TYPE field identifies the ICMP message, the CODE field provides further information about the associated TYPE field, and the CHECKSUM provides a method for determining the integrity of the message.

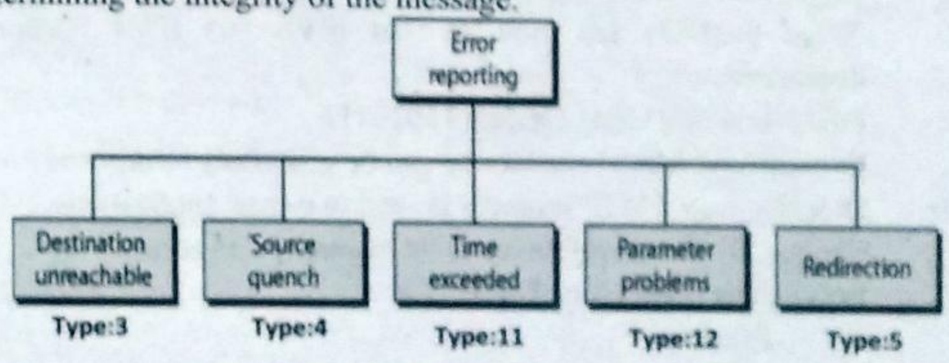



Figure: Some Error types

Scanned by TapScanner

# Model Questions Sets For Practice

## MODEL SET 1

Bachelor Level/Second Year/ Fourth Semester/ Science — Full Marks: 60
**Computer Networks (CSC 258)** — Pass Marks: 40
Time: 3 hours

*Candidates are required to give their answer in their own words as far as practicable.*

The figures in the margin indicate full marks.

**Group A (Long Answer question Section)**

**Attempt any TWO questions.** **[2×10=20]**

1. Explain the reason behind layering the network architecture? Write the functions of all layers of TCP/IP model.
2. What do you mean by routing? Differentiate between Non-adaptive algorithm and adaptive algorithm.
3. Suppose we want to transmit the message 1011 0010 0111 and protect it from errors using the CRC polynomial $x^4+x^2+1$. Use polynomial long division to determine the message that should be transmitted. Suppose the leftmost bit of the message is inverted due to noise on the transmission link. What is the result of the receiver's CRC calculation? How does the receiver known that an error has occurred?

**Group B (Long Answer question Section)**

**Attempt any EIGHT questions.** **[8×5=40]**

4. What is transmission media? Explain about any three transmission media in detail.
5. Is 192.16.30.64/26 a host, network or broadcast address? Also write subnet mask and usable hosts of the following network.
6. Explain how a TCP connection can be gracefully terminated?
7. What methods are used so that IPV6 and IPV4 networks are interoperable?
8. Find Hamming Code for data 11000011.
9. Explain how a TCP connection can be gracefully terminated?
10. Describe how SMTP protocol is used in e-mail applications.
11. Explain DNS with reference to its hierarchy and records.
12. Write short notes (Any Two):
    a. Ethernet
    b. CDMA
    c. Single bit parity



## MODEL SET 2

Bachelor Level/Second Year/ Fourth Semester/ Science — Full Marks: 60
**Computer Networks (CSC 258)** — Pass Marks: 40
Time: 3 hours.

*Candidates are required to give their answer in their own words as far as practicable.*

The figures in the margin indicate full marks.

**Group A (Long Answer question Section)**

**Attempt any TWO questions.** **[2×10=20]**

1. Why do we need layered protocol architecture? Discuss each layer of TCP/IP architecture along with function of each layer. Compare TCP/IP with OSI mode.
2. Assume a Class B network. Divide this network in 16 different sub-networks. What is new subnet mask?
3. Define multiplexing. What are the benefits of using multiplexing? Discuss Go-Back-N and Selective Repeat protocol in detail.

**Group B (Long Solution question Section)**

**Attempt any EIGHT questions.** **[8×5=40]**

4. What is circuit-switching? How is it different from packet-switching?
5. Discuss different services provided by DNS.
6. What are the causes of congestion? Discuss leaky bucket algorithm.
7. Discuss different fields of UDP. How is it different from TCP?
8. What is static routing? How is it different from dynamic routing?
9. What is random access protocol? Discuss FDMA in detail.
10. Differentiate between thicknet and thinnet cable.
11. What is network management? Discuss different areas of network management.
12. Write short notes on:
    a. Star Topology
    b. SNMP

Scanned by TapScanner

## MODEL SET 3

Bachelor Level/Second Year/ Fourth Semester/ Science — Full Marks: 60

**Computer Networks (CSC 258)** — Pass Marks: 40

Time: 3 hours.

*Candidates are required to give their answer in their own words as far as practicable.*

The figures in the margin indicate full marks.

**Group A (Long Answer question Section)**

**Attempt any TWO questions.** **[2×10=20]**

1. Explain the principles of application layer protocols. What do you mean by file transfer?
2. What are the main relationship between transport layer and network layer? What are the transport layer uses in Internet?
3. Explain Dijkastra shortest path algorithm. Also show working of Dijkastra algorithm with the help of an example.

**Group B (Long Answer question Section)**

**Attempt any EIGHT questions.** **[8×5=40]**

4. Describe in detail about lightwave transmission.
5. Explain the working principle of DNS.
6. What do you mean by pipelined reliable data transfer protocol?
7. What do you mean by hierarchical routing?
8. Explain the multicasting routine and its applications.
9. Define data link layer and its services.
10. What are the responsibilities of Presentation layer and Session layer of OSI Model?
11. Explain the Asynchronous transfer mode (ATM).
12. Write short notes on:
    a. Packet Switching
    b. DHCP



## MODEL SET 4

Bachelor Level/Second Year/ Fourth Semester/ Science — Full Marks: 60

**Computer Networks (CSC 258)** — Pass Marks: 40

Time: 3 hours.

*Candidates are required to give their answer in their own words as far as practicable.*

The figures in the margin indicate full marks.

**Group A (Long Answer question Section)**

**Attempt any TWO questions.** **[2×10=20]**

1. What is the significance of layered architecture? Explain the OSI layered architecture with neat sketch.
2. With an example explain the Dynamic routing algorithm used in computer networks.
3. Define DNS. Explain the DNS records and DNS messages.

**Group B (Long Answer question Section)**

**Attempt any EIGHT questions.** **[8×5=40]**

4. What do you mean by Internet Protocol Stack?
5. Differentiate between transport layer and network layer.
6. Explain the principle of congestion control.
7. What do you mean by IP datagram fragmentation?
8. Explain Give a detail note on slotted ALOHA protocols.
9. List the functions of bridge? How does a router differ from a bridge?
10. Explain about GBN Sliding window protocol.
11. Explain the term checksum in reference to error detection codes.
12. Write short notes on:
    a. Bluetooth
    b. Leaky Bucket

Scanned by TapScanner

## MODEL SET 5

Bachelor Level/Second Year/ Fourth Semester/ Science — Full Marks: 60

**Computer Networks (CSC 258)** — Pass Marks: 40

Time: 3 hours.

*Candidates are required to give their answer in their own words as far as practicable.*

The figures in the margin indicate full marks.

**Group A (Long Answer question Section)**

**Attempt any TWO questions.** **[2×10=20]**

1. What are the seven layers of OSI model? Explain the structure of TCP Header format.
2. What do you mean by routing? Different between Non-adaptive algorithm and adaptive algorithm.
3. Consider 8 bit 10101010 is transmitted using Hamming distance. Show the actual bits transmitted without any error. If the 6th bit from LSB is inverted during transmission. Show how the error is detected and corrected at the receiver's end. Also, obtain the original data after correction.

**Group B (Long Answer question Section)**

**Attempt any EIGHT questions.** **[8×5=40]**

4. What do you mean by Internet Protocol Stack?
5. Differentiate between DNS records and DNS messages
6. Explain the pipelined reliable data transfer protocol.
7. Explain network service model.
8. Explain IPv4 addressing.
9. Explain on ALOHA and Slotted ALOHA protocols.
10. What are the various applications of multimedia networking?
11. What are the various types of multimedia networking applications?
12. Write short notes on:
    a. IMAP
    b. SCTP



## MODEL SET 6

Bachelor Level/Second Year/ Fourth Semester/ Science — Full Marks: 60

**Computer Networks (CSC 258)** — Pass Marks: 40

Time: 3 hours.

*Candidates are required to give their answer in their own words as far as practicable.*

The figures in the margin indicate full marks.

**Group A (Long Answer question Section)**

**Attempt any TWO questions.** **[2×10=20]**

1. Explain the functioning of 7 layers of OSI model. What is the necessary of using 7 layers concept in OSI model?
2. Explain the various field of the TCP header and the working of the TCP protocol.
3. Explain how does CRC detect the error with multiple bits? Given message is $M(x) = x^7+x^4+x^3+x^2+1$ and the generator is $G(x) = x3 + 1$. Show the actual bit string transmitted, suppose the third bit from the left is inverted during the transmission. Show how the error is detected at the receiver's end.

**Group B (Long Answer question Section)**

**Attempt any EIGHT questions.** **[8×5=40]**

4. What are sliding window protocol? Explain one -bit sliding window protocol with an appropriate diagram.
5. Explain how slotted ALOHA improves the performance of system over pure ALOHA.
6. Why routing is important in a computer? Differentiate between adaptive and non-adaptive routing algorithms.
7. Differentiate between broadband and base band services.
8. How does ATM differ from frame relay? List and briefly define the ATM service classes.
9. Compare and contrast the IPv4 and the IPv6 header files. Do they have any fields in common?
10. Define multiplexing. Discuss the need for multiplexing in network system.
11. What is meant by "domain name"? How is a domain name translated to an equivalent IP address? Explain
12. Write short notes on:
    a. NFV
    b. NATing

Scanned by TapScanner

## MODEL SET 7

Bachelor Level/Second Year/ Fourth Semester/ Science — Full Marks: 60

**Computer Networks (CSC 258)** — Pass Marks: 40

Time: 3 hours.

*Candidates are required to give their answer in their own words as far as practicable.*

The figures in the margin indicate full marks.

**Group A (Long Answer question Section)**

**Attempt any TWO questions.** **[2×10=20]**

1. Define Protocol. Why do we need layered protocol architecture? Discuss each layer of TCP/IP protocol architecture in detail
2. Define transmission media. Differentiate between guided and unguided transmission media. Discuss each guided transmission media in detail.
3. What is routing? Discuss link state routing algorithm in detail.

**Group B (Long Answer question Section)**

**Attempt any EIGHT questions.** **[8×5=40]**

4. Explain client server system. How is it different from peer to peer system?
5. Explain the final delivery of email to the end user using POP3.
6. Discuss the importance of multiplexing in data communication.
7. Assume a class A network and divide it into four subnets. What is the value of new subnet mask?
8. Discuss CRC as an error detection mechanism.
9. Illustrate the frame structure of IEEE 802.3
10. What is congestion control? Why do we need it?
11. Differentiate between circuit switching, packet switching and message switching.
12. Write short notes on:
    a. DNS
    b. Category 3 UTP and Category 5 UTP



## MODEL SET 8

Bachelor Level/Second Year/ Fourth Semester/ Science — Full Marks: 60

**Computer Networks (CSC 258)** — Pass Marks: 40

Time: 3 hours.

*Candidates are required to give their answer in their own words as far as practicable.*

The figures in the margin indicate full marks.

**Group A (Long Answer question Section)**

**Attempt any TWO questions.** **[2×10=20]**

1. Discuss the relationship between transport layer and network layer. Discuss TCP as a transport layer protocol along with its segment structure.
2. What is the significance of ISDN? Explain the basic concept of ISDN. Give the protocol architecture of ISDN?
3. Why do we need routing algorithm? Discuss distance vector routing algorithm in detail.

**Group B (Long Answer question Section)**

**Attempt any EIGHT questions.** **[8×5=40]**

4. Which layer implements the node to node channel connection in OSI layered architecture?
5. Discuss FTP in detail.
6. Why do we need multiplexing in data communication? Discuss.
7. What is subnetting? Assume a class C network and divide it into four subnets. What is the value of new subnet mask?
8. How does the system correct error after error detection?
9. What is transmission media? Discuss wireless transmission media in detail.
10. Suppose you have got 50 PCs to set up in a LAN, which cable, device, protocol, you prefer and why justify with reason.
11. Explain leaky bucket and token bucket algorithms.
12. Write short notes on:
    a. HTTP
    b. Backbone

Scanned by TapScanner

## MODEL SET 9

Bachelor Level/Second Year/Fourth Semester/Science — Full Marks: 60

**Computer Networks (CSC 258)** — Pass Marks: 48

Time: 3 hours

*Candidates are required to give their answer in their own words as far as practicable.*

The figures in the margin indicate full marks.

**Group A (Long Answer question Section)**

**Attempt any TWO questions.** [2×10=20]

1. Explain the seven layer of OSI model and compare between them.
2. What do you mean by 'link state routing algorithm'? Differentiate between IPv4 and IPv6.
3. Explain the purpose of subnetting and also explain the subnet mask.

**Group B (Long Answer question Section)**

**Attempt any EIGHT questions.** [8×5=40]

4. Explain the responsibilities of data link layer in the internet model.
5. What do you mean by TCP push operation?
6. Explain the Internet Control Message Protocol (ICMP) along with the function.
7. What are the differences between 10base2 and 10base5 cables?
8. What is the need for error detection? Explain with typical examples.
9. Differentiate between ALOHA and slotted ALOHA protocols.
10. What is the use of SNMP protocol in a network?
11. What are the groups of HTTP header? Tabulate the various HTTP request operations.
12. Write short notes on:
    a. NGN
    b. Frame Relay



## MODEL SET 10

Bachelor Level/Second Year/Fourth Semester/Science — Full Marks: 60

**Computer Networks (CSC 258)** — Pass Marks: 48

Time: 3 hours.

*Candidates are required to give their answer in their own words as far as practicable.*

The figures in the margin indicate full marks.

**Group A (Long Answer question Section)**

**Attempt any TWO questions.** [2×10=20]

1. What is protocol? Why do we need layered protocol architecture? Explain each layer of OSI protocol architecture in detail.
2. How does transport layer support flow control? Discuss TCP as a transport layer protocol along with its segment structure.
3. Discuss the importance of routing algorithm. Discuss distance vector routing algorithm. Compare it with link state routing.

**Group B (Long Answer question Section)**

**Attempt any EIGHT questions.** [8×5=40]

4. Explain connectionless service. How is it different from connection oriented service?
5. How does multiplexing increase transmission efficiency? Explain.
6. Explain the message transfer using Simple Mail Transfer Protocol.
7. Assume a class C network and divide it into eight subnets. What is the value of new subnet mask?
8. Explain Hamming Code as error detection and correction mechanism with suitable example.
9. Discuss the infrastructure of network management.
10. Mention the type of multimedia networking applications.
11. What is piggybacking? Explain in detail along with suitable example.
12. Write short notes on:
    a. ISDN
    b. RIP

□□□

Scanned by TapScanner

# Database Management System

Syllabus

Course Title: Database Management System
Course No: CSC260
Nature of the Course: Theory + Lab
Semester: IV

Full Marks: 60 + 20 + 20
Pass Marks: 24 + 8 + 8
Credit Hrs: 3

**Course Description:**

The course covers the basic concepts of databases, database system concepts and architecture, data modeling using ER diagram, relational model, SQL, relational algebra and calculus, normalization, transaction processing, concurrency control, and database recovery.

**Course Objective:**

The main objective of this course is to introduce the basic concepts of database, data modeling techniques using entity relationship diagram, relational algebra and calculus, basic and advanced features SQL, normalization, transaction processing, concurrency control, and recovery techniques.

**Course Contents:**

**Unit 1: Database and Database Users** **(2 Hrs.)**

Introduction; Characteristics of the Database Approach; Actors on the Scene; Workers behind the Scene; Advantages of Using the DBMS Approach

**Unit 2: Database System – Concepts and Architecture** **(3 Hrs.)**

Data Models, Schemas, and Instances; Three-Schema Architecture and Data Independence; Database Languages and Interfaces; the Database System Environment; Centralized and Client/Server Architectures for DBMSs; Classification of Database Management Systems

**Unit 3: Data Modeling Using the Entity-Relational Model** **(6 Hrs.)**

Using High-Level Conceptual Data Models for Database Design; Entity Types, Entity Sets, Attributes, and Keys; Relationship Types, Relationship Sets, Roles, and Structural Constraints; Weak Entity Types; ER Diagrams, Naming Conventions, and Design Issues; Relationship Types of Degree Higher Than Two; Subclasses, Superclasses, and Inheritance; Specialization and Generalization; Constraints and Characteristics of Specialization and Generalization

**Unit 4: The Relational Data Model and Relational Database Constraints** **(3 Hrs.)**

Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations, Transactions, and Dealing with Constraint Violations

**Unit 5: The Relational Algebra and Relational Calculus** **(5 Hrs.)**

Unary Relational Operations: SELECT and PROJECT; Relational Algebra Operations from Set Theory; Binary Relational Operations: JOIN and DIVISION; Additional Relational Operations; the Tuple Relational Calculus; the Domain Relational Calculus



**Unit 6: SQL** **(8 Hrs.)**

Data Definition and Data Types; Specifying Constraints; Basic Retrieval Queries; Complex Retrieval Queries; INSERT, DELETE, and UPDATE Statements; Views

**Unit 7: Relational Database Design** **(7 Hrs.)**

Relational Database Design Using ER-to-Relational Mapping; Informal Design Guidelines for Relational Schemas; Functional Dependencies; Normal Forms Based on Primary Keys; General Definitions of Second and Third Normal Forms; Boyce-Codd Normal Form; Multivalued Dependency and Fourth Normal Form; Properties of Relational Decomposition

**Unit 8: Introduction to Transaction Processing Concepts and Theory** **(4 Hrs.)**

Introduction to Transaction Processing; Transaction and System Concepts; Desirable Properties of Transactions; Characterizing Schedules Based on Recoverability; Characterizing Schedules Based on Serializability

**Unit 9: Concurrency Control Techniques** **(4 Hrs.)**

Two-Phase Locking Technique; Timestamp Ordering; Multiversion Concurrency Control; Validation (Optimistic) Techniques and Snapshot Isolation Concurrency Control

**Unit 10: Database Recovery Techniques** **(3 Hrs.)**

Recovery Concepts; NO-UNDO/REDO Recovery Based on Deferred Update; Recovery Technique Based on Immediate Update; Shadow Paging; Database Backup and Recovery from Catastrophic Failures

**Laboratory Works:**

The laboratory work includes writing database programs to create and query databases using basic and advanced features of structured query language (SQL).

**Text Books:**

1. Fundamentals of Database Systems; Seventh Edition; Ramez Elmasri, Shamkant B. Navathe; Pearson Education
2. Database System Concepts; Sixth Edition; Avi Silberschatz, Henry F Korth, S Sudarshan; McGraw-Hill

**Reference Books:**

1. Database Management Systems; Third Edition; Raghu Ramakrishnan, Johannes Gehrke; McGraw-Hill
2. A First Course in Database Systems; Jaffrey D. Ullman, Jennifer Widom; Third Edition; Pearson Education Limited

Scanned by TapScanner

TRIBHUVAN UNIVERSITY

Institution of Science and Technology

Course Title: Database Management System — Full Marks: 60

Course No: CSC260 — Pass Marks: 24

Semester: IV — Time: 3 Hrs

## T.U. MODEL QUESTIONS-ANSWERS SET I

**Section A (Long questions)**

**Attempt any two questions.** **(2 × 10 = 20)**

**1. Consider the following database and write SQL as given:** **(5 × 2 = 10)**

**Customer (Cno, Cname, Caddress, Ccontact)**

**Purchase (Cno, Pid)**

**Product (Pid, Pname, price, quantity)**

**Answer:**

**a. Find the names of all products having price 1000.**

```
SELECT Pname
FROM Product
WHERE price=1000;
```

**b. Find the name of those customers who purchased Dell Laptop.**

```
SELECT Cname
FROM Customer NATURAL JOIN Purchase NATURAL
JOIN Product
WHERE Pname="Dell Laptop";
```

**c. Find the total number of products purchased by customer 'Ram'**

```
SELECT COUNT (Pid)
FROM Customer NATURAL JOIN Purchase
WHERE Cname="Ram";
```

**d. Increase price of all products by 5 %**

```
UPDATE Product
SET price=price + price*0.05;
```

**e. Find total price of Apple Mobiles**

```
SELECT SUM (price)
FROM Product
WHERE Pname="Apple Mobiles";
```



**2. What are the benefits of using normalization? Discuss 1NF, 2NF, and 3NF with suitable example.** **(2.5 + 7.5 = 10)**

**Answer:** Here we can see why normalization is an attractive prospect in RDBMS concepts.

- A smaller database can be maintained as normalization eliminates the duplicate data. Overall size of the database is reduced as a result.
- Better performance is ensured. As databases become lesser in size, the passes through the data becomes faster and shorter thereby improving response time and speed.
- Narrower tables are possible as normalized tables will be fine-tuned and will have lesser columns which allows for more data records per page.
- Fewer indexes per table ensure faster maintenance tasks (index rebuilds).
- Also realizes the option of joining only the tables that are needed.

**First Normal Form (1NF)**

A relation will be 1NF if it contains an atomic value. It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute. First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

**Example:** Relation EMPLOYEE is not in 1NF because of multi-valued attribute EMP_PHONE.

EMPLOYEE table

| EMP_ID | EMP_NAME | EMP_PHONE | EMP_STATE |
|---|---|---|---|
| 14 | John | 7272826385, 9064738238 | State I |
| 20 | Harry | 8574783832 | State II |
| 12 | Aarav | 7390372389, 8589830302 | State IV |

The decomposition of the EMPLOYEE table into 1NF has been shown below:

| EMP_ID | EMP_NAME | EMP_PHONE | EMP_STATE |
|---|---|---|---|
| 14 | John | 7272826385 | State I |
| 14 | John | 9064738238 | State I |
| 20 | Harry | 8574783832 | State II |
| 12 | Aarav | 7390372389 | State IV |
| 12 | Aarav | 8589830302 | State IV |

Scanned by TapScanner

**Second Normal Form (2NF)**

In the 2NF, relational must be in 1NF. In the second normal form, all non-key attributes are fully functional dependent on the primary key

**Example**: Let's assume, a school can store the data of teachers and the subjects they teach. In a school, a teacher can teach more than one subject.

TEACHER table

| TEACHER_ID | SUBJECT | TEACHER_AGE |
|---|---|---|
| 25 | Chemistry | 30 |
| 25 | Biology | 30 |
| 47 | English | 35 |
| 83 | Math | 38 |
| 83 | Computer | 38 |

In the given table, non-prime attribute TEACHER_AGE is dependent on TEACHER_ID which is a proper subset of a candidate key. That's why it violates the rule for 2NF. To convert the given table into 2NF, we decompose it into two tables:

TEACHER_DETAIL table:

| TEACHER_ID | TEACHER_AGE |
|---|---|
| 25 | 30 |
| 47 | 35 |
| 83 | 38 |

TEACHER_SUBJECT table:

| TEACHER_ID | SUBJECT |
|---|---|
| 25 | Chemistry |
| 25 | Biology |
| 47 | English |
| 83 | Math |
| 83 | Computer |

**Third Normal Form (3NF)**

A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency. 3NF is used to reduce the data duplication. It is also used to achiev the data integrity. If there is no transitive dependency for no prime attributes, then the relation must be in third normal form.

A relation is in third normal form if it holds at least one of the following conditions for every non-trivial function dependency $X \rightarrow Y$.



- X is a super key.
- Y is a prime attribute, i.e., each element of Y is part of some candidate key.

**Example:**

EMPLOYEE_DETAIL table:

| EMP_ID | EMP_NAME | EMP_ZIP | EMP_STATE | EMP_CITY |
|---|---|---|---|---|
| 222 | Geeta | 201010 | State I | Dhankuta |
| 333 | Manmohan | 02228 | State IV | Butwal |
| 444 | Laxmi | 60007 | State IV | Dang |
| 555 | Janaki | 06389 | State II | Janakpur |
| 666 | Aarav | 462007 | State VI | Nepalgang |

**Super key in the table above**: {EMP_ID},
{EMP_ID, EMP_NAME}, {EMP_ID, EMP_NAME, EMP_ZIP}
....so on

**Candidate key:** {EMP_ID}

**Non-prime attributes:** In the given table, all attributes except EMP_ID are non-prime.

Here, EMP_STATE & EMP_CITY dependent on EMP_ZIP and EMP_ZIP dependent on EMP_ID. The non-prime attributes (EMP_STATE, EMP_CITY) transitively dependent on super key (EMP_ID). It violates the rule of third normal form. That's why we need to move the EMP_CITY and EMP_STATE to the new <EMPLOYEE_ZIP> table, with EMP_ZIP as a Primary key.

EMPLOYEE table:

| EMP_ID | EMP_NAME | EMP_ZIP |
|---|---|---|
| 222 | Geeta | 201010 |
| 333 | Manmohan | 02228 |
| 444 | Laxmi | 60007 |
| 555 | Janaki | 06389 |
| 666 | Aarav | 462007 |

EMPLOYEE_ZIP table:

| EMP_ZIP | EMP_STATE | EMP_CITY |
|---|---|---|
| 201010 | State I | Dhankuta |
| 02228 | State IV | Butwal |
| 60007 | State IV | Dang |
| 06389 | State II | Janakpur |
| 462007 | State VI | Nepalgang |

**3. Define Relational Algebra (RA) and explain its six fundamental operations with suitable example. (2 + 8 = 10)**

**Answer:** Relational algebra, first created by Edgar F. Codd while at IBM, is a family of algebras with a well-founded semantics used for modeling the data stored in relational databases, and defining queries on it.

Relational algebra is a procedural query language. It gives a step by step process to obtain the result of the query. It uses operators to perform queries.

Scanned by TapScanner

Six fundamental operations or RA are listed below:

- Select operation
- Project operation
- Union operation
- Set difference operation
- Cartesian product operation
- Rename operation

**Select operation**

The SELECT operation (denoted by σ (sigma)) is used to select a subset of the tuples from a relation based on a selection condition.

- The selection condition acts as a filter
- Keeps only those tuples that satisfy the qualifying condition
- Tuples satisfying the condition are selected whereas the other tuples are discarded (filtered out)

**Syntax:** $\sigma_{<condition>}$ (Relation)

**Selection Example**

Assume the following relation Employee has the following tuples:

Employee

| Name | Office | Dept | Rank |
|---|---|---|---|
| Bipin | 400 | Computer | Assistant |
| Niky | 220 | Economics | Adjunct |
| Rahul | 160 | Economics | Assistant |
| Binita | 420 | Computer | Associate |
| Solu | 500 | Finance | Associate |

Select only those Employees who involve in the Computer department.

$\sigma_{Dept = 'Computer'}$ (Employee)

**Result:**

| Name | Office | Dept | Rank |
|---|---|---|---|
| Bipin | 400 | CS | Assistant |
| Binita | 420 | CS | Associate |

**Project operation**

Projection operation is used to extracts specified columns (arity) of a relation. With the help of this operation, any number of columns can be omitted from a table or columns of table can rearrange.

**Syntax:** $\Pi_{<attribute\text{-}list>}$ (Relation)

Rorito Lyk



**Projection Examples**

Assume the same Employee relation above is used. Project only the names and departments of the employees:

$\Pi_{name, dept}$ (Employee)

**Results**

| Name | Dept |
|---|---|
| Bipin | Computer |
| Niky | Economics |
| Rahul | Economics |
| Binita | Computer |
| Solu | Finance |

**Union operation**

Consider the following relations **R** and **S**. The **union** of relations R and S is denoted by R U S and it is the set of tuples that are either in R or in S or in both. It returns the union (set union) of two compatible relations. For a union operation to be legal, we require that invoked relations must have the same number of attributes and corresponding attributes have same type.

**R**

| First | Last | Age |
|---|---|---|
| Bill | Smith | 22 |
| Kamala | dhami | 21 |
| Maya | Singh | 23 |
| Anisha | Jha | 22 |

**S**

| First | Last | Age |
|---|---|---|
| Pinky | ojha | 36 |
| Maya | singh | 23 |
| Anisha | KC | 22 |

**Result:** Relation with tuples from R and S with duplicates removed.

| First | Last | Age |
|---|---|---|
| Bill | Smith | 22 |
| kamala | dhami | 21 |
| Maya | singh | 23 |
| Anisha | Jha | 22 |
| Pinky | Ojha | 36 |
| Anisha | KC | 22 |

**Set difference operation**

Set difference is denoted by the minus sign (-). It finds tuples that are in one relation, but not in another. Thus results in a relation containing tuples that are in **R** but not in **S**.

**Result:** Result: Relation with tuples from R but not from S

**R - S**

| First | Last | Age |
|---|---|---|
| Bill | Smith | 22 |
| Maya | Singh | 23 |
| Anisha | Jha | 22 |

Scanned by TapScanner

**Cartesian product operation**

The Cartesian product operation does not require relations to union-compatible i.e. the involved relations may have different schemas. The **Cartesian product** of two relations R and S is denoted by R **X** S, is the set of all possible combinations of tuples of the two relations.

**Example:**

R

| First | Last | Age |
|---|---|---|
| Kamala | Ojha | 22 |
| Pawan | Bhatt | 23 |
| Anisha | KC | 32 |

S

| Dinner | Dessert |
|---|---|
| Steak | Ice Cream |
| Lobster | Cheesecake |

**Result:** Produce all combinations of tuples from two relations.

R X S

| First | Last | Age | Dinner | Dessert |
|---|---|---|---|---|
| Kamala | Ojha | 22 | Steak | Ice Cream |
| Kamala | Ojha | 22 | Lobster | Cheesecake |
| Pawan | Bhatt | 23 | Steak | Ice Cream |
| Pawan | Bhatt | 23 | Lobster | Cheesecake |
| Anisha | KC | 32 | Steak | Ice Cream |
| Anisha | CK | 32 | Lobster | Cheesecake |

**Rename operation**

The rename operator is denoted by rho ( ρ ). It can be used in two ways:

- $\rho_x(E)$ Return the result of expression E in the table named *x*.
- $\rho_x (A_1, A_2, \ldots, A_n) (E)$ Return the result of expression E in the table named *x* with the attributes renamed to $A_1, A_2, \ldots, A_n$.

It is mainly used in the situation where we need to find the Cartesian product of a relation with itself i.e. Account × Account.

For that we should rename one of the account tables by some other name to avoid the confusion.

**Example:** $\rho_{Emp\ (Name1,\ Dept1)}$ (Employee)

**Employee**

| Name | Department |
|---|---|
| Bhupi | IT |
| Arjun | CSC |
| Aayan | IT |

After rename →

**Emp**

| Name1 | Dept1 |
|---|---|
| Bhupi | IT |
| Arjun | CSC |
| Aayan | IT |

Rorito Lyk



## Section B (Short questions)

**Attempt any eight questions.** **(8 × 5 = 40)**

**4. What database schema? What are functions of database administrator?** **2 +3 = 5)**

**Answer:** The database schema of a database system is its structure described in a formal language supported by the database management system (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases).

**Student**

| Name | Student-number | Class | Major |
|---|---|---|---|

Fig: Schema diagram for Student

**Functions of database administrator are listed below:**

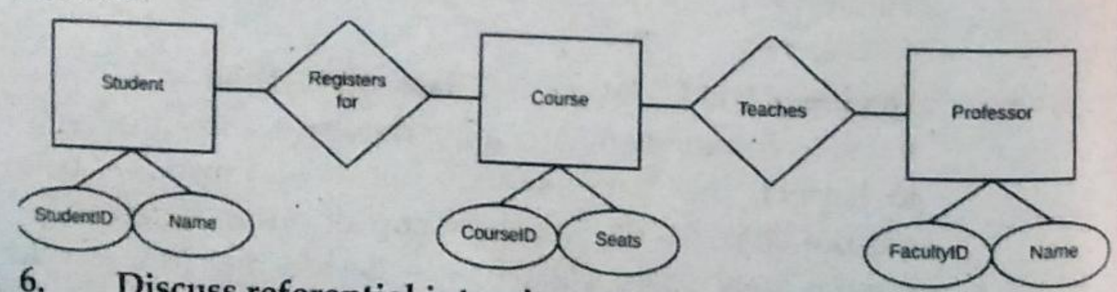
- **Installing and Configuration of database**
  DBA is responsible for installing the database software. He/She configures the software of database and then upgrades it if needed. There are many database software like oracle, Microsoft SQL and MySQL in the industry so DBA decides how the installing and configuring of these database software will take place.
- **Deciding the hardware device**
  Depending upon the cost, performance and efficiency of the hardware, it is DBA who have the duty of deciding which hardware devise will suit the company requirement. It is hardware that is an interface between end users and database so it needed to be of best quality.
- **Managing Data Integrity**
  Data integrity should be managed accurately because it protects the data from unauthorized use. DBA manages relationship between the data to maintain data consistency.
- **Decides Data Recovery and Back up method**
  If any company is having a big database, then it is likely to happen that database may fail at any instance. It is require that a DBA takes backup of entire database in regular time span. DBA has to decide that how much data should be backed up and how frequently the back should be taken. Also the recovery of data base is done by DBA if they have lost the database.

Scanned by TapScanner

- **Tuning Database Performance**
  Database performance plays an important role for any business. If user is not able to fetch data speedily then it may loss company business. So by tuning a modifying SQL commands a DBA can improves the performance of database.
- **Capacity Issues**
  All the databases have their limits of storing data in it and the physical memory also has some limitations. DBA has to decide the limit and capacity of database and all the issues related to it.
- **Database design / Physical organization modification**
  The logical design of the database is designed by the DBA. Also a DBA is responsible for physical design, external model design, and integrity control. The DBA modifies the physical organization of the database to reflex the changing needs of the organization or to improve performance.
- **Database accessibility**
  DBA writes subschema to decide the accessibility of database. He decides the users of the database and also which data is to be used by which user. No user has to power to access the entire database without the permission of DBA.
- **Decides validation checks on data**
  DBA has to decide which data should be used and what kind of data is accurate for the company. So he always puts validation checks on data to make it more accurate and consistence.

5. **Construct an E-R diagram for online course registration where students register courses online.** (5)

**Answer:**




6. **Discuss referential integrity with example.** (5)

**Answer:** In relational model we use primary key and foreign key to establish relationships between two relations. Table containing primary key attribute is called master table and the table

Rorito Lyk



containing foreign key attribute is called child table. **Referential integrity** ensures that value appeared in foreign key attribute of child table must also appear in primary key attribute of corresponding master table. Defining referential integrity restricts database modification operations such as insert, delete, and update. To illustrate this consider following two relations:

**Employee**

| Eid | Ename | Salary | DNO |
|---|---|---|---|
| E01 | Ramesh | 26000 | D01 |
| E02 | Sohan | 19000 | D01 |
| E03 | Renuka | 25000 | D02 |
| E04 | Dina | 22000 | D03 |

**Department**

| DNO | Dname | Location |
|---|---|---|
| D01 | IT | Gwarko |
| D02 | Finance | Satdobato |
| D03 | HR | Balkumari |

**Insert:** We cannot insert new tuples containing value of foreign key attribute that do not appear in primary key attribute of related table. For example, we cannot insert new employee that works in D04 department because the department D04 does not exists in department table.

**Delete**: We cannot delete tuples containing values of primary key attribute that also appear foreign key attribute of related table. For example, we cannot delete tuple containing value D02 of DNO from department table because there are employees working in department D02.

**Update:** There are two cases of update operation

**Case 1:** We cannot change the value of primary key attribute if the child table contains related values. For example, we cannot change value of DNO from D01 to D05 in department table because employee table contains employees working in department D01.

**Case 2:** We cannot change value of foreign key attribute without modifying value of corresponding primary key attribute. For example, we cannot change value of DNO from D01 to D05 in department table.

Thus for referential integrity a foreign key can have only two possible values- either the relevant primary key or a null value. No other values are allowed.

7. **What is functional dependency? Why do we need inference rules?** (2 + 3 = 5)

**Answer:** Functional dependency plays a key role in differentiating good database design from bad database design. Functional dependency is a relationship that exists when one attribute uniquely determines another attribute. If R is a relation with

Scanned by TapScanner

attributes X and Y, a functional dependency between th attributes is represented as X→Y, which specifies Y functionally dependent on X. Here X is a determinant set and is a dependent attribute. Each value of X is associated wit precisely one Y value.

Functional dependency in a database serves as a constrain between two sets of attributes. Defining functional dependenc is an important part of relational database design an contributes to aspect normalization.

The Armstrong's axioms are the basic inference rul Armstrong's axioms are used to conclude functiona dependencies on a relational database. The inference rule is type of assertion. It can apply to a set of FD (functiona dependency) to derive other FD. Using the inference rule, w can derive additional functional dependency from the initi set. The Functional dependency has 6 types of inference rule:

**Reflexivity**

If Y ⊆ X then X→Y. The axiom of reflexivity indicates that give a set of attributes the set itself functionally determines any of it own subsets.

**Augmentation**

If X→Y and Z is a subset of table R (i.e., Z is any set of attribute in R), then XZ→YZ. The axiom of augmentation indicates th we can augment the left side of the functional dependency o both sides conveniently with one or more attributes. The axion does not allow augmenting the right-hand side alone. Th augmentation rule can be represented as If X→Y then XZ→Y

**Transitivity**

If X→Y and Y→Z then X→Z. The axiom of transitivity indicate that if one attribute uniquely determines a second attribute an this, in turn, uniquely determines a third one, then the firs attributes determines the third one. Consider three parallel line X, Y, and Z. The line X is parallel to line Y. The line Y is paralle to line Z then it implies that line X is parallel to line Z. Th property is called transitivity.

**Pseudo transitivity**

If X→Y and YW→Z then XW→Z. Transitivity is a special cas of pseudo transitivity when W is null. The axiom of pseud transitivity is a generalization of the transitivity axiom.

**Union**

If X→Y and X→Z then X→YZ. The axiom of union indicate that if there are two functional dependencies with the sam



determinant it is possible to form a new functional dependency that preserves the determinant and has its right-hand side the union of the right-hand sides of the two functional dependencies.

**Decomposition**

If X→YZ then X→Y and X→Z. The axiom of decomposition indicates that the determinant of any functional dependency can uniquely determine any individual attribute or any combination of attributes of the right-hand side of the functional dependency.

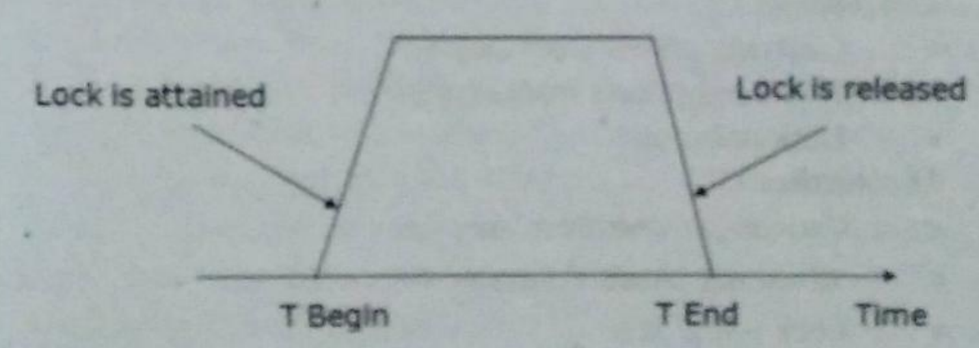
**8. Why do we need concurrency control? Discuss two phase locking protocol. (2+3=5)**

**Answer:** Concurrency control is the procedure in DBMS for managing simultaneous operations without conflicting with each another. Concurrent access is quite easy if all users are just reading data. There is no way they can interfere with one another. Though for any practical database, would have a mix of reading and WRITE operations and hence the concurrency is a challenge.

Concurrency control is used to address such conflicts which mostly occur with a multi-user system. It helps you to make sure that database transactions are performed concurrently without violating the data integrity of respective databases. Therefore, concurrency control is a most important element for the proper functioning of a system where two or multiple database transactions that require access to the same data, are executed simultaneously.

**Two-phase locking (2PL)**

The two-phase locking protocol divides the execution phase of the transaction into three parts. In the first part, when the execution of the transaction starts, it seeks permission for the lock it requires. In the second part, the transaction acquires all the locks. The third phase is started as soon as the transaction releases its first lock. In the third phase, the transaction cannot demand any new locks. It only releases the acquired locks.




Scanned by TapScanner

There are two phases of 2PL:

- **Growing phase:** In the growing phase, a new lock on the data item may be acquired by the transaction, but none can be released.
- **Shrinking phase:** In the shrinking phase, existing locks held by the transaction may be released, but no new locks can be acquired.

In the below example, if lock conversion is allowed then the following phase can happen:

- Upgrading of lock (from S(a) to X (a)) is allowed in growing phase.
- Downgrading of lock (from X(a) to S(a)) must be done in shrinking phase.

**Example:**

| | T1 | T2 |
|---|---|---|
| 0 | LOCK-S(A) | |
| 1 | | LOCK-S(A) |
| 2 | LOCK-X(B) | |
| 3 | — | — |
| 4 | UNLOCK(A) | |
| 5 | | LOCK-X(C) |
| 6 | UNLOCK(B) | |
| 7 | | UNLOCK(A) |
| 8 | | UNLOCK(C) |
| 9 | — | — |

The following way shows how unlocking and locking work with 2-PL.

**Transaction T1:**

- Growing phase: from step 1-3
- Shrinking phase: from step 5-7
- Lock point: at 3

**Transaction T2:**

- Growing phase: from step 2-6
- Shrinking phase: from step 8-9
- Lock point: at 6

Rorito Lyk



**9. Why do we need database recovery? Discuss shadow paging technique for database recovery. (2 + 3 = 5)**

**Answer:** A computer system, like any other device, is subject to failure from a variety of causes: disk crash, power outage, software error, a fire in the machine room, even sabotage. In any failure, information may be lost. Therefore, the database system must take actions in advance to ensure that the atomicity and durability properties of transactions are preserved. An integral part of a database system is a recovery scheme that can restore the database to the consistent state that existed before the failure. The recovery scheme must provide high availability; that is, it must minimize the time for which the database is not usable after a failure

**Shadow Paging**

Shadow paging is an alternative to log-based recovery techniques, which has both advantages and disadvantages. It may require fewer disk accesses, but it is hard to extend paging to allow multiple concurrent transactions. The paging is very similar to paging schemes used by the operating system for memory management. The idea is to maintain two page tables during the life of a transaction: the current page table and the shadow page table. When the transaction starts, both tables are identical. The shadow page is never changed during the life of the transaction. The current page is updated with each write operation. Each table entry points to a page on the disk. When the transaction is committed, the shadow page entry becomes a copy of the current page table entry and the disk block with the old data is released. If the shadow is stored in nonvolatile memory and a system crash occurs, then the shadow page table is copied to the current page table. To recover from a failure during transaction execution, it is sufficient to free the modified database pages and to discard the current directory. The state of the database before transaction execution is available through the shadow directory, and that state is recovered by simply copying page table into current page table. The database thus is returned to its state prior to the transaction that was executing when the crash occurred.

This guarantees that the shadow page table will point to the database pages corresponding to the state of the database prior to any transaction that was active at the time of the crash, making aborts automatic.

Scanned by TapScanner

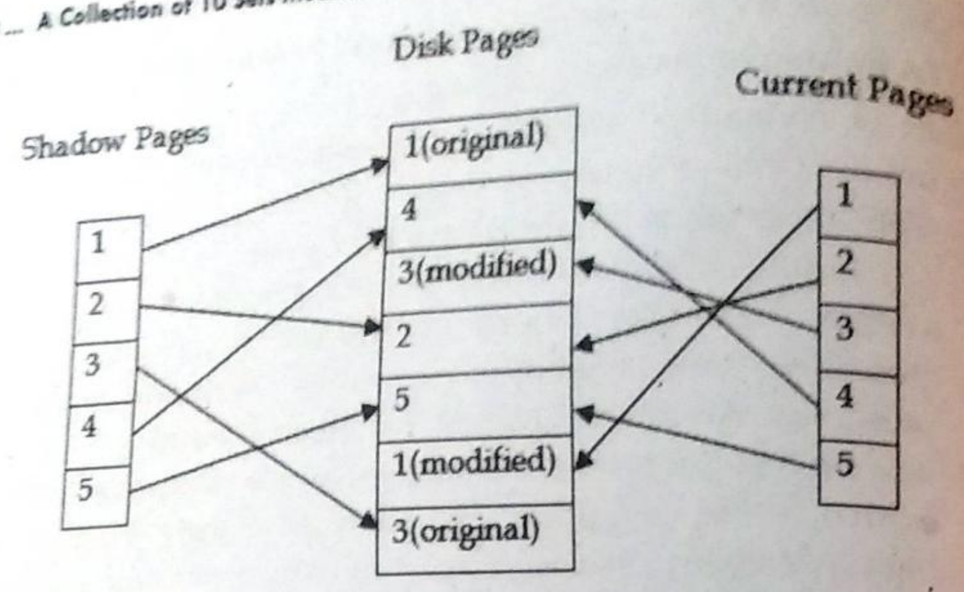



Figure: Shadow paging

Advantages

- No Overhead for writing log records.
- No Undo / No Redo algorithm.
- Recovery is faster.

Disadvantages

- Data gets fragmented or scattered.
- After every transaction completion database pages containing old version of modified data need to be garbage collected.
- Hard to extend algorithm to allow transaction to run concurrently.

**10. Differentiate concept of Centralized and Client/Server Architectures for DBMSs with suitable example. (5)**

**Answer: Centralized database system**

In a centralized system, all programs run on the main host computer, including the DBMS, the application that accesses the database and the communication facilities that send and receive data from the user's terminals. The users access the database through either locally connected or dial-up (remote) terminals. The terminals are generally dumb, having little or no processing power of their own and consists of only a screen, keyboard and hardware to communicate with the host.

Rorito Lyk-yk



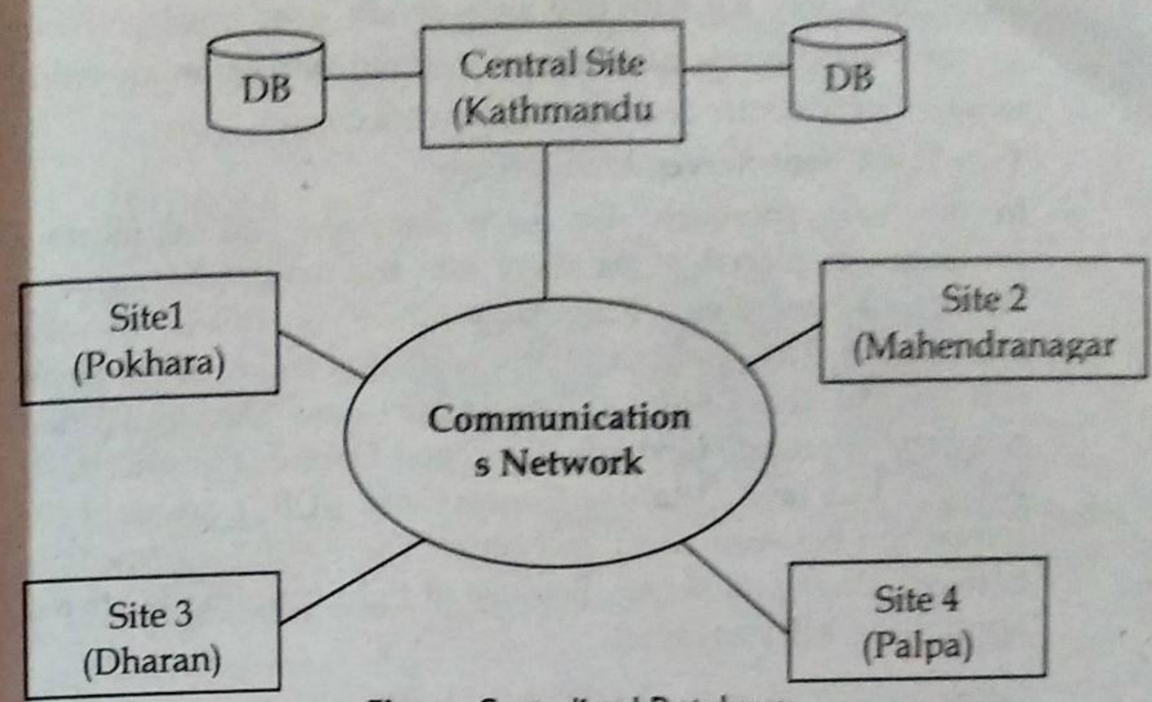



**Figure:** Centralized Database

**Client/Server database system**

In a generalized concept, client PC is the computer from where the user requests for data and information and the server provides the requested information. The database application on the client PC referred to as the 'front end system' athat handles all the screen and user input/output processing.

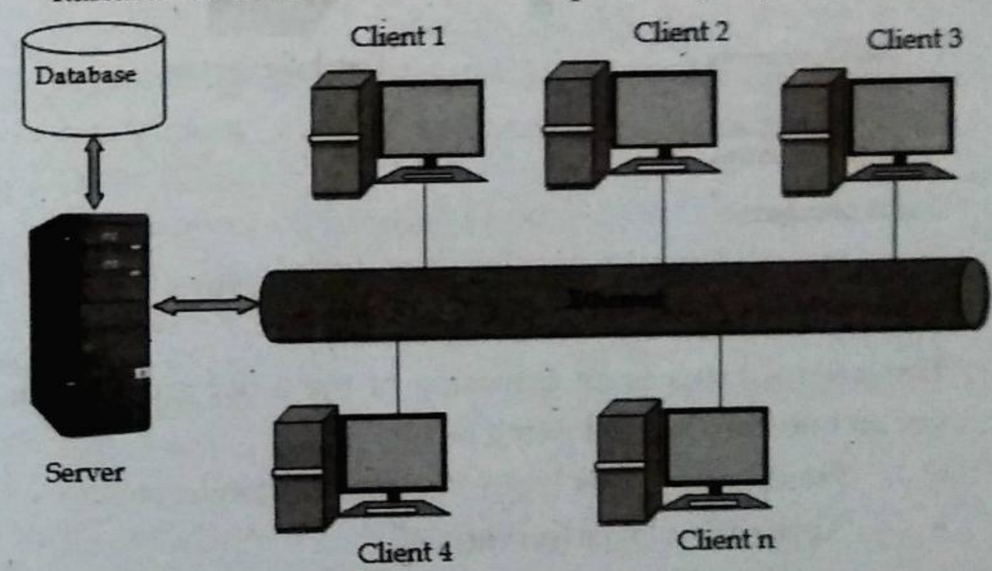



**Figure:** Client server database system

The 'back end system' on the database server handles data processing and disk access. For example a user on the front end creates a query for data from the database server and the front-end application sends the request across the network to the server. The database server performs the actual search and

Scanned by TapScanner

sends back only the data that answers the user's query. There are two approaches to implement client/server architecture: two-tier architecture and three-tier architecture.

a. **Two Tier Client-Server Architecture**

In the first approach, the user interface and application programs are placed on the client side and the database system on the server side. This architecture is called two-tier architecture. The application programs that reside at the client side invoke the DBMS at the server side. The application program interface standards like Open Database Connectivity (ODBC) and Java Database Connectivity (JDBC) are used for interaction between client and server.There is no intermediate between client and server. Because of tight coupling a 2 tiered application will run faster.

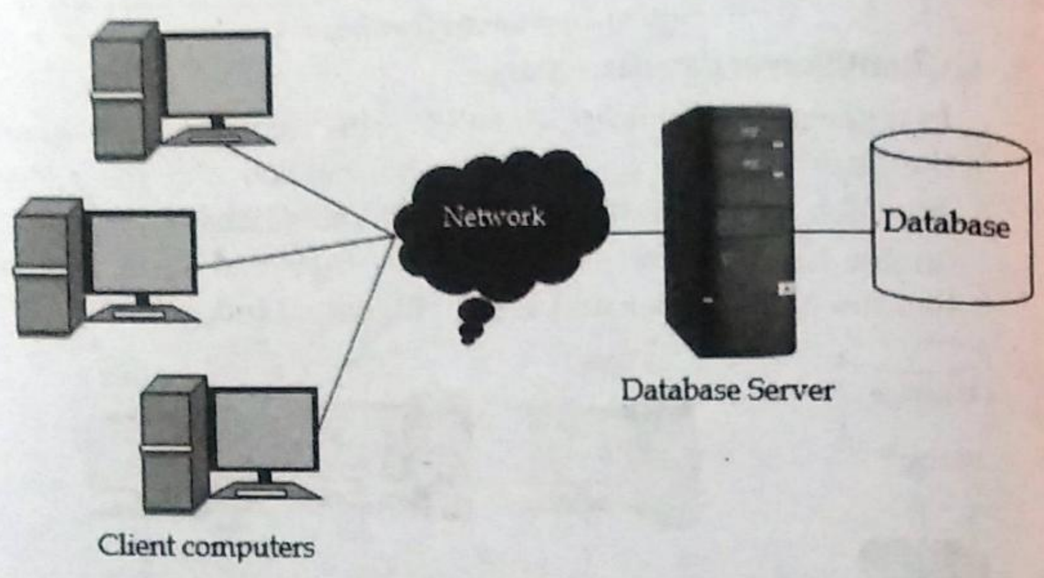



Figure: Two-tier client/server architecture

b. **Three Tier Client-Server Architecture**

Three-tier schema is an extension of the 2-tier architecture. 3-tier architecture has following layers

- Presentation layer (your PC, Tablet, Mobile, etc.)
- Application layer (server)
- Database Server

This DBMS architecture contains an Application layer between the user and the DBMS, which is responsible for communicating the user's request to the DBMS system and send the response from the DBMS to the user. The application layer (business logic layer) also processes functional logic,

Rorito Lyk



constraint, and rules before passing data to the user or down to the DBMS. Three tier architecture is the most popular DBMS architecture.

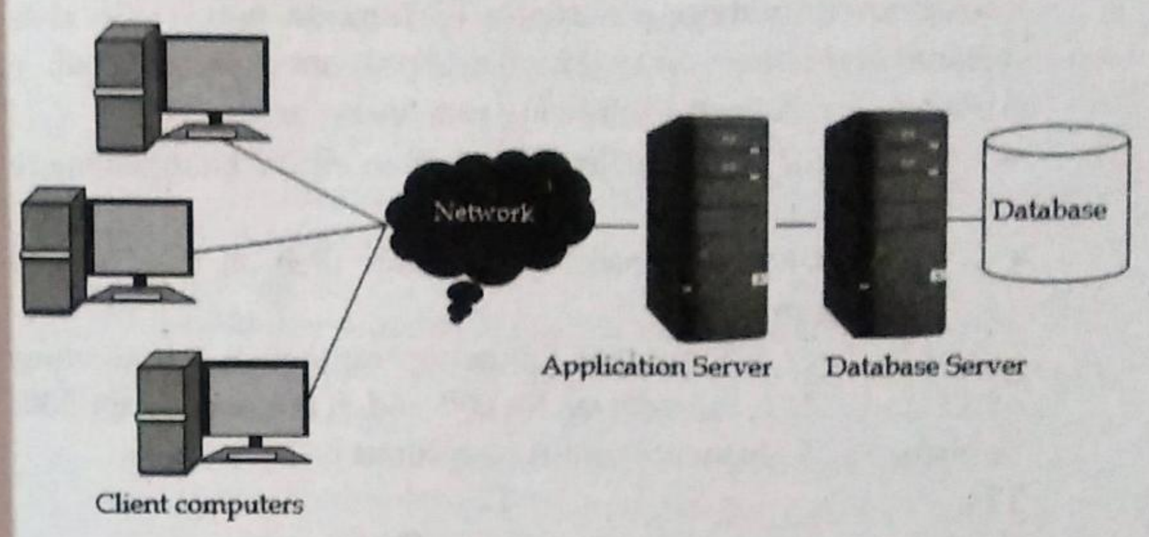



Figure: three-tier clien/server architecture

**11. Define Transaction and explain its desirable properties. (5)**

**Answer:** The transaction is a set of logically related operation. It contains a group of tasks. A transaction is an action or series of actions. It is performed by a single user to perform operations for accessing the contents of the database. Example: Suppose an employee of bank transfers Rs 800 from X's account to Y's account. This small transaction contains several low-level tasks:

**X's Account**

Open_Account(X)
Old_Balance = X.balance
New_Balance = Old_Balance - 800
X.balance = New_Balance
Close_Account(X)

**Y's Account**

Open_Account(Y)
Old_Balance = Y.balance
New_Balance = Old_Balance + 800
Y.balance = New_Balance
Close_Account(Y)

**Property of Transaction**

- Atomicity
- Consistency
- Isolation
- Durability

Scanned by TapScanner

**Atomicity**

It states that all operations of the transaction take place at once if not, the transaction is aborted. There is no midway, i.e., the transaction cannot occur partially. Each transaction is treated as one unit and either run to completion or is not executed at all.

Atomicity involves the following two operations:

- **Abort:** If a transaction aborts then all the changes made are not visible.
- **Commit:** If a transaction commits then all the changes made are visible.

**Example:** Let's assume that following transaction T consisting of T1 and T2. A consists of Rs 600 and B consists of Rs 300. Transfer Rs 100 from account A to account B.

| T1 | T2 |
|---|---|
| Read(A) | Read(B) |
| A:= A-100 | B:= B+100 |
| Write(A) | Write(B) |

After completion of the transaction, A consists of Rs 500 and B consists of Rs 400. If the transaction T fails after the completion of transaction T1 but before completion of transaction T2, then the amount will be deducted from A but not added to B. This shows the inconsistent database state. In order to ensure correctness of database state, the transaction must be executed in entirety.

**Consistency**

The integrity constraints are maintained so that the database is consistent before and after the transaction. The execution of a transaction will leave a database in either its prior stable state or a new stable state. The consistent property of database states that every transaction sees a consistent database instance. The transaction is used to transform the database from one consistent state to another consistent state.

For example: The total amount must be maintained before or after the transaction.

- Total before T occurs = 600+300=900
- Total after T occurs= 500+400=900

Therefore, the database is consistent. In the case when T1 is completed but T2 fails, then inconsistency will occur.

Rorito Lyk



**Isolation**

It shows that the data which is used at the time of execution of a transaction cannot be used by the second transaction until the first one is completed. In isolation, if the transaction T1 is being executed and using the data item X, then that data item can't be accessed by any other transaction T2 until the transaction T1 ends. The concurrency control subsystem of the DBMS enforced the isolation property.

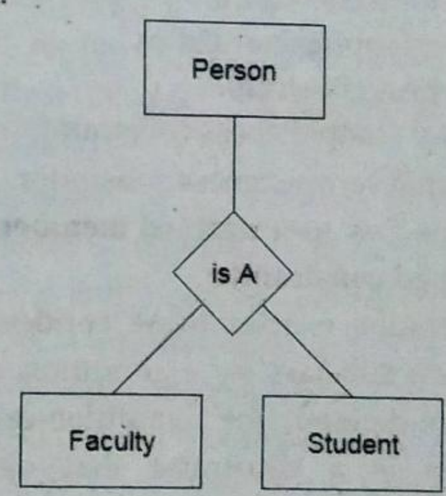
**Durability**

The durability property is used to indicate the performance of the database's consistent state. It states that the transaction made the permanent changes. They cannot be lost by the erroneous operation of a faulty transaction or by the system failure. When a transaction is completed, then the database reaches a state known as the consistent state. That consistent state cannot be lost, even in the event of a system's failure. The recovery subsystem of the DBMS has the responsibility of Durability property.

**12. Explain constraints and characteristics of Specialization and Generalization of data model. (5)**

**Answer: Generalization**

Generalization is like a bottom-up approach in which two or more entities of lower level combine to form a higher level entity if they have some attributes in common. In generalization, an entity of a higher level can also combine with the entities of the lower level to form a further higher level entity. Generalization is more like subclass and superclass system, but the only difference is the approach. Generalization uses the bottom-up approach. In generalization, entities are combined to form a more generalized entity, i.e., subclasses are combined to make a superclass. For example, Faculty and Student entities can be generalized and create a higher level entity Person.

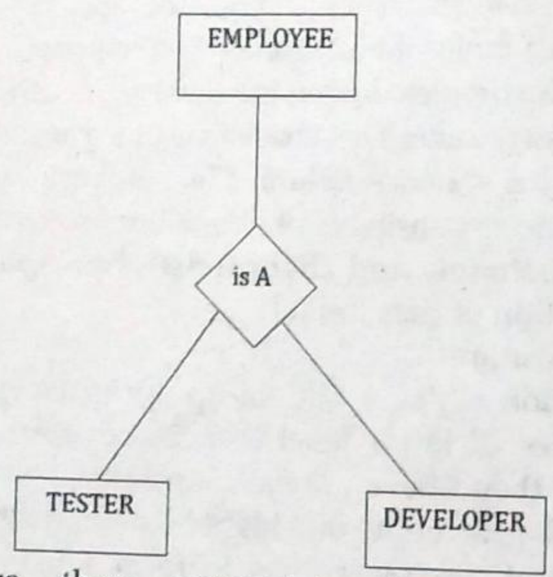



Scanned by TapScanner

### Specialization

Specialization is a top-down approach, and it is opposite to Generalization. In specialization, one higher level entity can be broken down into two lower level entities.

Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics. Normally, the superclass is defined first, the subclass and its related attributes are defined next, and relationship set are then added. **For example:** In an Employee management system, EMPLOYEE entity can be specialized as TESTER or DEVELOPER based on what role they play in the company.




There are three constraints that may apply to a specialization/generalization:

1. Membership constraints
   - Condition defined membership constraint
   - User defined membership constraint
2. Disjoint constraints
   - Disjoint constraint
   - Overlapping constraint
3. Completeness constraints
   - Total completeness constraint
   - Partial completeness constraint

**Condition defined vs. user defined membership constraint**

**Condition defined constraints**

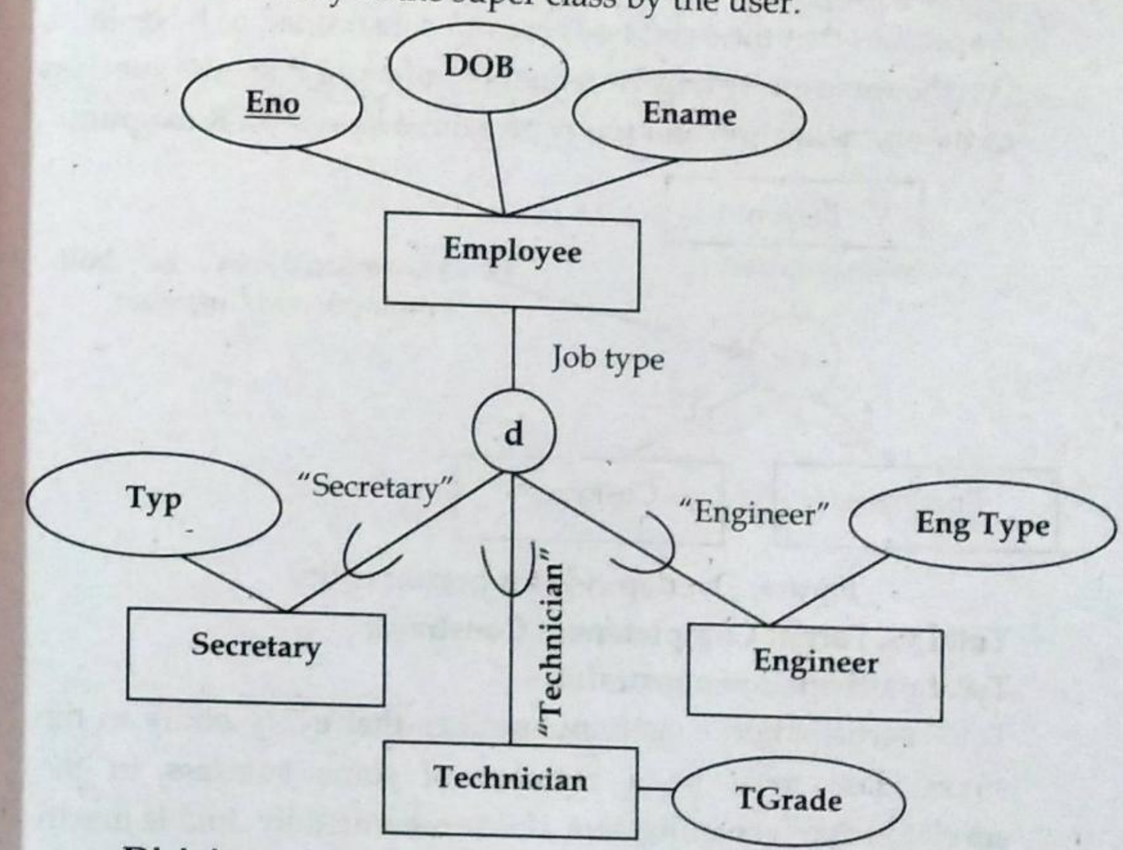
If we can determine exactly those entities that will become members of each subclass by a condition, the subclasses are called predicate-defined (or condition-defined) subclasses. Here, condition is a constraint that determines subclass



members. For example, if the employee entity set has an attribute job-type, we can specify the condition of membership in the secretary subclass by a condition (job-type="secretary"), which we call defining predicate of the subclass.

**User-defined constraints**

If no condition determines membership, the subclass is called user-defined. Membership in a subclass is determined by the database users by applying an operation to add an entity to the subclass. Membership in the subclass is specified individually for each entity in the super class by the user.




**Disjoint vs. overlapping constraint**

Another type of constraint relates to whether or not entities may belong to more than one lower-level entity set within a single generalization.

**Disjoint Constraint**

It specifies that the subclasses of the specialization must be disjoint. Here an entity can be a member of at most one of the subclasses of the specialization and it is represented by d in EER diagram. Simply, if an entity can be a member of at most one of the subclasses of the specialization, then the subclasses are called disjoint.

Scanned by TapScanner

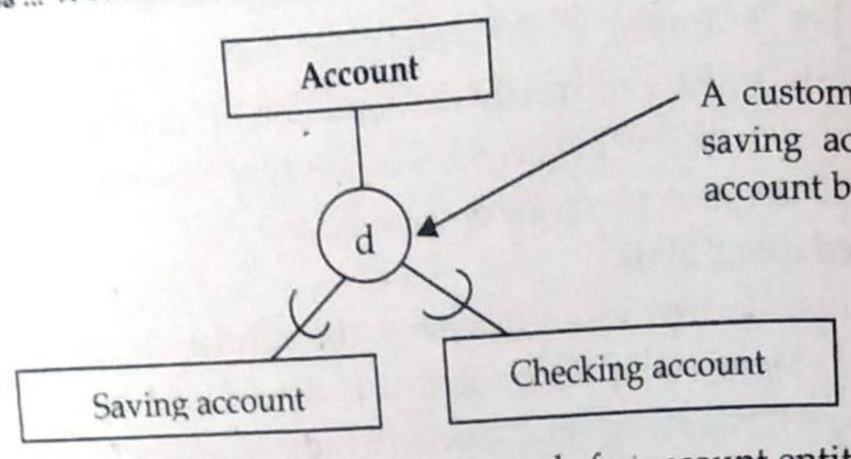



**Figure**: disjoint rule for account entity

**Overlapping constraint**

It specifies that the subclasses are not constrained to be disjoint, i.e., the same entity may be a member of more than one subclass of the specialization and it is represented by o in EER diagram.

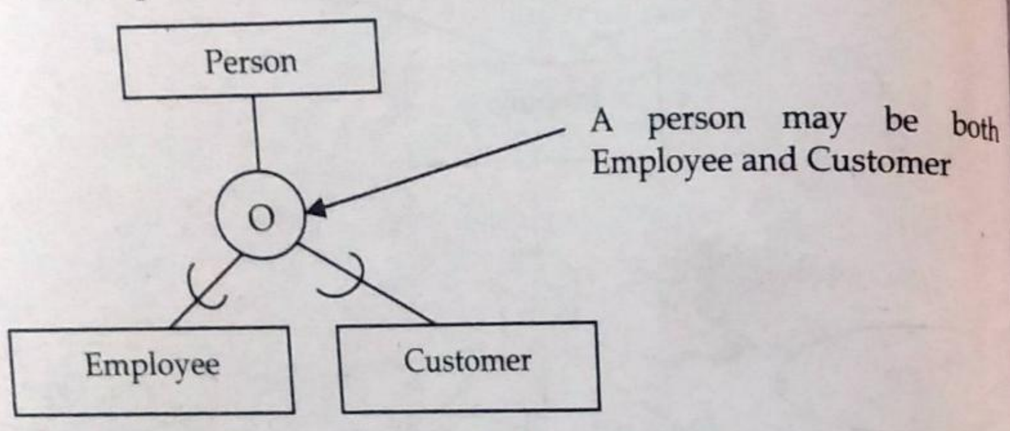



**Figure**: Overlap rule for person entity

**Total vs. Partial Completeness Constraint**

**Total participation constraint**

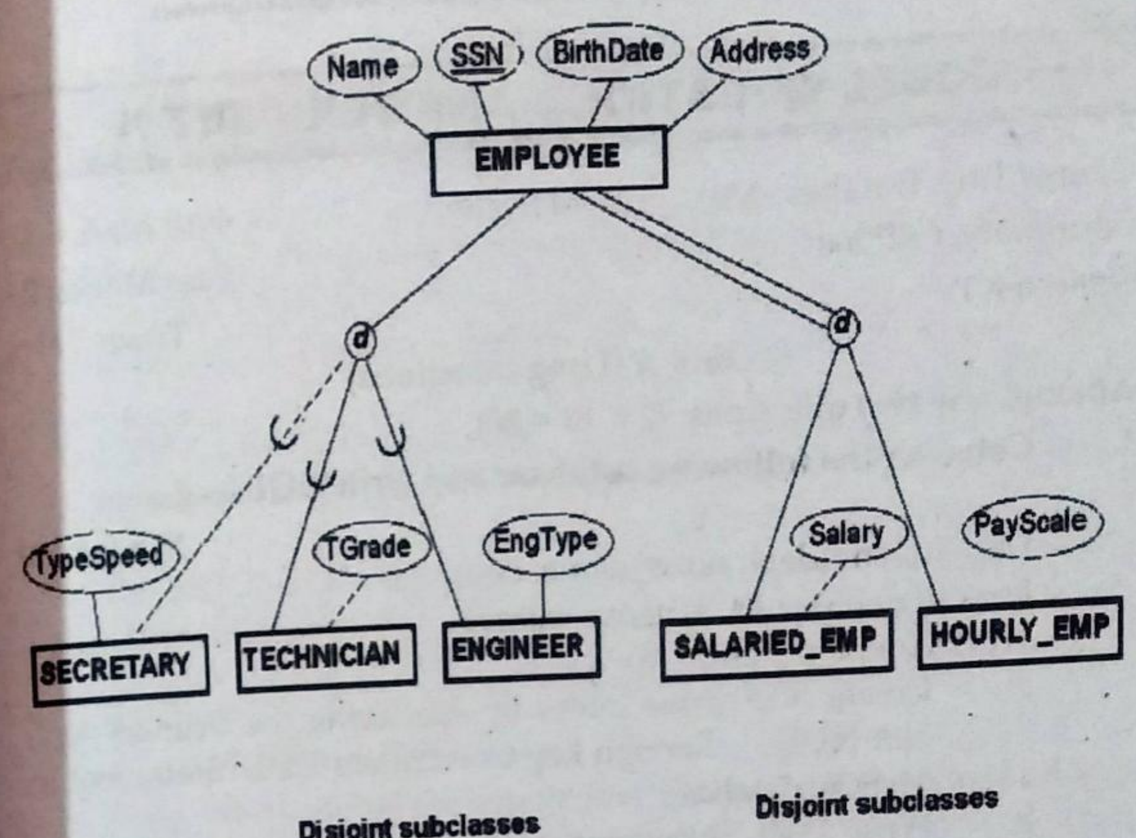
Total participation constraint specifies that every entity in the super class must be a member of some subclass in the specialization/generalization. It is represented by double line in EER diagram.

**Partial participation constraint**

It allows every entity in the super class may not belong to a subclass and shown in EER diagrams by a single line.

**Example**: if some Employee entities, (for example, sales representatives) do not belong to any of the subclasses {Secretary, Engineer, Technician}, then the specialization is partial.






Rorito Lyk

Scanned by TapScanner

## MODEL QUESTIONS-ANSWERS SET II

Course Title: Database Management System — Full Marks: 6
Course No: CSC260 — Pass Marks: 2
Semester: IV — Time: 3 Hr

Section A (Long questions)

**Attempt any two questions. (2 × 10 = 20)**

**1. Consider the following database and write SQL as given: (5 × 2 = 10)**

Department (dept_no, d_name, city)
Employee (emp_Id, e_name, salary)
Works (dept_no, emp_Id)

a. Create the above tables by specifying the Primary key, Not NULL , Foreign key Constraints DDL Statement in MySQL database
b. Write DML Statement to Insert any one record in each tables
c. Display the name of the employees.
d. Find the name of the employees whose salary is greater than 10000.
e. Find the department (d_name) of the employee 'Binek'.

**Answer:**

a. Create the above tables by specifying the Primary key, Not NULL , Foreign key Constraints DDL Statement in MySQL database

```
CREATE TABLE Departmant
(
Dept_no          INT PRIMARY KEY,
D_name           VARCHAR (20),
City             VARCHAR (20)
);
CREATE TABLE Employee
(
Emp_id           INT PRIMARY KEY,
e_name           VARCHAR (20),
Salary           FLOAT
);
CREATE TABLE Works
(
Dept_no          INT,
```



```
Emp_id           INT,
FOREIGN KEY (Dept_no) REFERENCES Department (Dept_no),
FOREIGN KEY (Emp_id) REFERENCES Employee (Emp_id)
);
```

b. Write DML Statement to Insert any one record in each tables

```
INSERT INTO Employee
VALUES (1, "Ramu", 5500);

INSERT INTO Departmant
VALUES (3, "Computer", "Kathmandu");

INSERT INTO Works
VALUES (1, 3);
```

c. Display the name of the employees.

```
SELECT e_name
FROM Employee;
```

d. Find the name of the employees whose salary is greater than 10000.

```
SELECT e_name
FROM Employee
WHERE salary > 10000;
```

e. Find the department (d_name) of the employee 'Binek'.

```
SELECT d_name
FROM Department NATURAL JOIN Works NATURAL JOIN
Employee
WHERE e_name = "Binek";
```

**2. What is normalization? Describe their importance in database. Compare BCNF and 3 NF with suitable example . (2+ 2+ 6 = 10)**

**Answer:** When developing the schema of a relational database, one of the most important aspects to be taken into account is to ensure that the duplication is minimized. This is done for two purposes:

- Reducing the amount of storage needed to store the data.
- Avoiding unnecessary data conflicts that may creep in because of multiple copies of the same data getting stored.

Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and update anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

Scanned by TapScanner

Normalization is a database design technique which organizes tables in a manner that reduces redundancy and dependency of data. It divides larger tables to smaller tables and links them using relationships.

**Importance of normalization in database**

Normalization of tables can be defined as the process during which poorly designed tables are decomposed by breaking up their attributes into smaller tables that possess desired properties. Normalization process eliminates uncontrolled data redundancies, thus eliminating the data anomalies and the data integrity problems that are produced by such redundancies.

**Difference between 3NF and BCNF**

3NF is a normal form that is used in normalizing a database design to reduce the duplication of data and ensure referential integrity by ensuring that the entity is in second normal form and all the attributes in a table are determined only by the candidate keys of that relation and not by any non-prime attributes. BCNF, on the other hand, is a normal form used in database normalization which is a slightly stronger version of the 3NF. Thus, this is the main difference between 3NF and BCNF.

The table should be in 2nd normal form, and there shouldn't be any transitive dependencies to satisfy 3NF while the prime attributes of the table should not depend on the non-prime attributes of the table to satisfy BCNF. Hence, functionality is another difference between 3NF and BCNF.

3. **What is procedural query language? How it is differ from non-procedural query language? Explain Aggregate functions and generalized projection operations in RA with suitable example.** **(2 +1+ 7 = 10)**

**Answer: Procedural Query Language:** This is a formal way of accessing the database. The information regarding what has to be accessed and how it has to be accessed are provided along with the queries, so that the data can be accessed from the database.

Relational Algebra is a type of Procedural Query Language. It consists of a set of operators which take one or two relations as input and a new relation is provided as the output. RA is the component of relational query engine. RA also provides a framework for query optimization. SQL queries are internally translated into RA operations first.



**Non Procedural Query Language:** This is an informal way of accessing the data from the database. In the case of Non Procedural Query Language, information is required only about what data has to be accessed from the database. SQL and Tuple Relational Calculus are types of Non Procedural Query Language.

**Aggregate functions**

Aggregate functions are algebraic functions that take a collection of values as input and return a single value as a result. It is denoted by symbol $\mathcal{G}$ read it as "calligraphic G".

General form of aggregate operation in relational algebra is;

$G_1, G_2 \ldots G_n\ \mathcal{G}_{F1(A1), F2(A2) \ldots Fn(An)}$ (E)

Where, E is any relational-algebra expression

$G_1, G_2 \ldots, G_n$ is a list of attributes on which to group (can be empty)

Each Fi is an aggregate function and each Ai is an attribute name.

All aggregate functions are deterministic. This means aggregate functions return the same value any time that they are called by using a specific set of input values. We cannot express aggregate functions by using basic relational algebra expressions. There are five aggregate functions that are included with most relational database systems. These operations are:

- SUM: sum of values
- AVG: average value
- MIN: minimum value
- MAX: maximum value
- COUNT: number of values

**Example**: Let's take an employee relation

**Employee**

| Eid | Name | Address | Age | Salary |
|---|---|---|---|---|
| $e_1$ | Rajan | Ktm | 33 | 34000 |
| $e_2$ | Aarav | Pokhara | 17 | 45000 |
| $e_3$ | Abin | Palpa | 22 | 55000 |
| $e_4$ | Ashna | Ktm | 19 | 24000 |
| $e_5$ | Manmohan | Pokhara | 30 | 33000 |

1. Find total number of employees

   $\mathcal{G}_{COUNT(Eid)}$ (Employee)

2. Find average age of employees of address 'Ktm'

   $\mathcal{G}_{AVG(Age)}$ ($\sigma_{Address="Ktm"}$ (Employee))

Scanned by TapScanner

**Generalized projection**

It extends the projection operation by allowing arithmetic functions to be used in the projection list. Generalized projection is enhanced version of project operation. It allows us to write aromatic operations containing attribute names and constants in projection list. General form of generalized projection is given below:

$\Pi_{F1, F2, F3, \ldots, Fn}$ (E)

Where, E is a relational algebra expression and Fi (i=1, 2....n) is an attribute or arithmetic expression containing attributes and constants.

**Example:** Let's take an employee relation

**Employee**

| eid | name | Age | Salary |
|---|---|---|---|
| $e_1$ | Rajan | 33 | 34000 |
| $e_2$ | Aarav | 17 | 45000 |
| $e_3$ | Abin | 22 | 55000 |
| $e_4$ | Ashna | 19 | 24000 |

1. Find name and salary of all employees by increasing their salary by 15%

   $\Pi_{name, salary=salary+salary*0.15}$ (Employee)

2. Increase the salary of all employees whose age greater than 20 by 5%.

   $\Pi_{eid, name, age, salary=salary+salary*0.05}$ ($\sigma_{age>20}$ (Employee))

**Section B (Short questions)**

**Attempt any eight questions.** **(8 × 5 = 40)**

4. **What is database user? Describe types of database users with example.** **(2 +3 = 5)**

**Answer:** The people who are accessing or working with the database are called database users and administrators. We know that the primary aim of the DBMS is to store the data or information and retrieve whenever it is needed by the database users. Those who are working with the database can be categorized into two: the database users and the database administrators.



The database users also can be categorized again into five groups according to how they interact with the database. They are:

- Native Users
- Application Programmers
- Sophisticated Users
- Specialized Users
- Stand-alone Users

**Native Users**

These are the database users who are communicating with the database through an already written program. For example: when a student is registering on a website for an online examination. He creates data in the database by entering and submitting his name, address and exam details.

**Application Programmers**

These are the software developers and programming professionals who write the program codes. They use tools like Rapid Application Development (RAD) tools for creating user interfaces with minimal efforts.

**Sophisticated Users**

Sophisticated users are those who are creating the database. These type of users do not write program code. And they do not use any software to request the database. The sophisticated users directly interact with the database system using query languages like SQL.

**Specialized Users**

The sophisticated users who write special database application programs are called specialized users. The write complex programs for the specific complex requirements. Specialized users write applications such as computer-aided design systems, knowledge-base and expert systems that store data having complex data types.

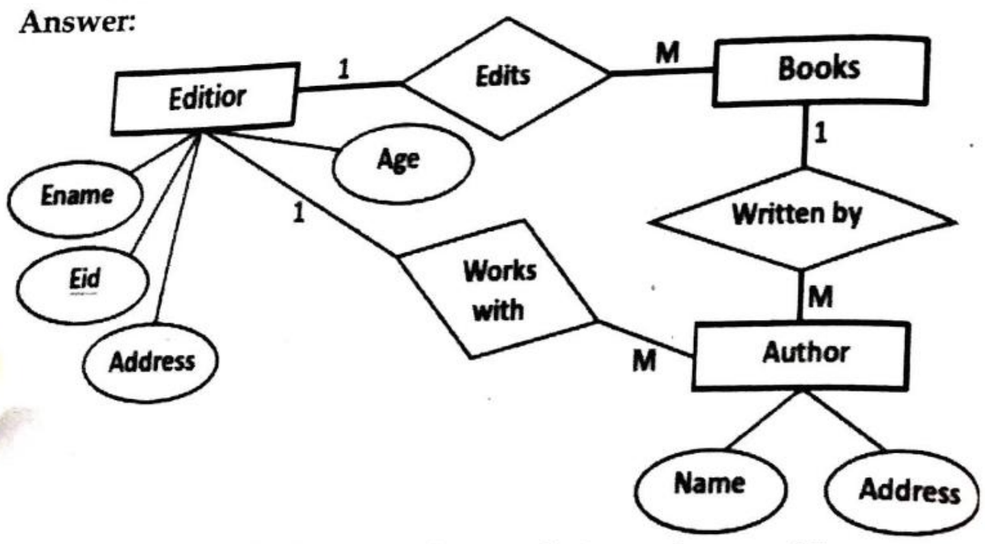
**Stand-alone Users**

Those who are using database for personal usage is called stand-alone users. There are many database packages for this type database user.

Scanned by TapScanner

5. **Draw ER diagram from following scenario:**
Each editor edits any number of books and works with any number of authors. Each book is written by any number of editors. Each author authors and is edited by any number of editors. Each author writes any number of books and works with any number of editors. (5)

**Answer:**




6. **Discuss CHECK constraint and Assertions with suitable example.** (2+3=5)

**Answer: CHECK Constraints**

CHECK constraint is added to the declaration of the attribute. The CHECK constraint may use the name of the attribute or any other relation or attribute name may in a sub-query. Attribute value check is checked only when the value of the attribute is inserted or updated. CHECK constraints allow users to prohibit an operation on a table that would violate the constraint. It is a local constraint.

***Example:*** *let's create a student table with attributes student id, student name, age and address. If we need to allow only those students in the table whose age must be an integer range 20 to 45, we could use the CHECK constraint during the creation of table as below:*

```
CREATE TABLE Student
(
sid                          INTEGER,
sname VARCHAR(20),
age                          INTEGER,
PRIMARY KEY (sid),
CHECK (age>=20 AND age<=45)
)
```



In the above student table if we are trying to insert a new record as

```
INSERT INTO Student
VALUES (5, "Rajesh", 15);
```

We get insertion is rejected message since value of age attribute violated the check condition.

Assertions are general purpose checks that allow the enforcement of any condition over the entire database. Similar to CHECK but they are global Constraints. When an assertion is made, the system tests it for validity, and tests it again on every update that may violate the assertion. This testing may introduce a significant amount of overhead; hence assertions should be used with great care. An assertion in SQL takes the form:

Create assertion <assertion-name> check <predicate>

**Example:** the department id of manager relation is always not null since each manager works at least one department.

```
CREATE ASSERTION NoBench CHECK
(NOT EXISTS
    (SELECT * FROM MANAGER
            WHERE DeptId IS NULL));
```

Above assertion ensures that there is no manager who is not assigned any department at any time. Let's take a manager relation in which some records are inserted as

**Manager**

| Mid | Mname | Address | DeptId |
|---|---|---|---|
| M01 | Aayan | Pokhara | D11 |
| M02 | Bhupi | Lalitpur | D22 |
| M03 | Arjun | Kathmandu | D11 |
| M05 | Ramesh | Palpa | Null |

In the above table the department id of manager 'Ramesh' is null due to which assertion is violated and we cannot further modify the database.

Assertions can be dropped using the DROP ASSERTION command.

Scanned by TapScanner

7. What is fully and partial functional dependency? Discuss insertion anomalies of database design with suitable practical example. (2 + 3 = 5)

Answer: Fully Functional Dependency

An attribute is fully functionally dependent on a second attribute if and only if it is functionally dependent on the second attribute but not on any subset of the second attribute. Mathematically, for a relation schema R, in a functional dependency X→Y, Y is said to be fully functional dependent on X if Z→Y is false for all Z ⊂ X.

**Partial Functional Dependency**

An attribute is partial functionally dependent on a second attribute if and only if it is functionally dependent on the second attribute and also dependency occur on any subset of the second attribute. This is the situation that exists if it is necessary to only use a subset of the attributes of the composite determinant to identify its object. Mathematically, for a relation schema R, in a functional dependency X→Y, Y is said to be partial functional dependent on X if by removal of some attributes from X and the dependency still holds.

**Example:** The dependency {Emp-Id, Projecta-No} → {Emp-Name} is partial because Emp-Id→Emp-Name also holds.

**Insertion anomalies of database design**

If a table is not properly normalized and has data redundancy then it will not only eat up extra memory space but will also make it difficult to handle and update the database, without facing data loss. Insertion, updating and Deletion Anomalies are very frequent if database is not normalized. To understand these anomalies let us take an example of a Student table.

**Student**

| Sid | Sname | Branch | HOD | Office Phone |
|---|---|---|---|---|
| 100 | Alim | Pokhara branch | Mr. Nabin | 4434564 |
| 101 | Aarav | Pokhara branch | Mr. Nabin | 4434564 |
| 102 | Aafiana | Pokhara branch | Mr. Nabin | 4434564 |
| 103 | Geeta | Pokhara branch | Mr. Nabin | 4434564 |
| 104 | Umesh | Banepa branch | Mr. Prabin | 5212343 |
| 105 | Ramesh | Banepa branch | Mr. Prabin | 5212343 |



In the table above, we have data of 6 Computer Sci. students. As we can see, data for the fields branch, HOD and Office_Phone is repeated for the students who are in the same branch in the college, this is Data Redundancy.

**Insertion Anomaly**

Suppose for a new admission, until and unless a student opts for a branch, data of the student cannot be inserted, or else we will have to set the branch information as NULL.

Also, if we have to insert data of 100 students of same branch, then the branch information will be repeated for all those 100 students. These scenarios are nothing but Insertion anomalies.

**Update Anomaly**

What if Mr. Prabin leaves the college? Or is no longer the HOD of computer science department? In that case all the student records will have to be updated, and if by mistake we miss any record, it will lead to data inconsistency. This is Updating anomaly.

**Deletion Anomaly**

In our Student table, two different information are kept together, Student information and Branch information. Hence, at the end of the academic year, if student records are deleted, we will also lose the branch information. This is Deletion anomaly.

8. **What is serial and non-serial schedule? Discuss Serializability with suitable example.** **(2+3=5)**

Answer: A schedule in which transactions are aligned in such a way that one transaction is executed first. When the first transaction completes its cycle then next transaction is executed. Transactions are ordered one after other. This type of schedule is called serial schedule as transactions are executed in a serial manner. In a multi-transaction environment, serial schedules are considered as benchmark. In a serial schedule only one transaction is active at a time. The commit (or abort) of one transaction initiates execution of the next transaction. Every serial schedule is regarded as correct because every transaction is correct if executed on its own. So T1 followed by T2, is correct so is T2 followed by T1. Hence, it does not matter which transaction is executed first. Serial Schedule limit concurrency if one transaction waits for I/O operation to complete, the CPU

Scanned by TapScanner

cannot be switched to some other transaction. Hence serial schedules are considered unacceptable in practice.

A schedule where the operations from a set of concurrent transactions are interleaved is called non-serial schedule. Although non-serial schedules are important to achieve efficient throughput in multi-user environment, they do not produce correct result always. The execution sequence of instruction in a transaction cannot be changed but two transactions can have their instruction executed in random fashion. This execution does no harm if two transactions are mutually independent and working on different segment of data but in case these two transactions are working on same data, results may vary. This ever-varying result may cause the database in an inconsistent state. Example of serial and non-serial schedule is given below.

| T1 | T2 |
|---|---|
| Read(A) | |
| A= A-50 | |
| Write(A) | |
| Read(B) | |
| B=B+50 | |
| Write(B) | |
| | Read(A) |
| | Temp= A*0.1 |
| | A=A-Temp |
| | Write(A) |
| | Read(B) |
| | B= B + Temp |
| | Write(B) |

Serial schedule

| T1 | T2 |
|---|---|
| Read(A) | |
| A= A-50 | |
| Write(A) | |
| | Read(A) |
| | Temp= A*0.1 |
| | A=A-Temp |
| | Write(A) |
| Read(B) | |
| B=B+50 | |
| Write(B) | |
| | Read(B) |
| | B= B + Temp |
| | Write(B) |

Non-serial schedule

**Serializability**

A given non-serial schedule of n transactions is serializable if it is equivalent to some serial sched ' That is if a non-serial schedule produce the same result a ot the serial schedule then the given non-serial schedule is said to be serializable. A schedule that is not serializable is called a non-serializable. The main objective of serializability is to search non-serial schedules that allow transaction to execute concurrently without

Rorito Lyk



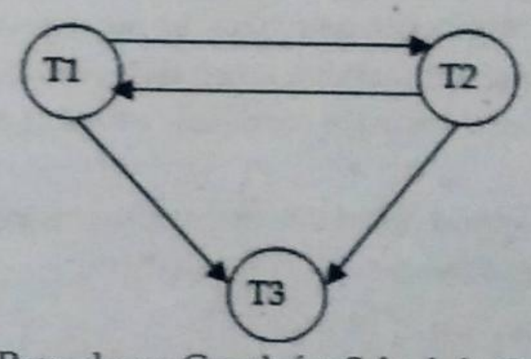
interfering one another transaction and produce the result that could be produced by a serial execution. In serializability, ordering of read/write is important. If two transactions only read data item they do not conflict and order is not important. If two transactions either read or write completely separate data items, they do not conflict and order is not important. If one transaction writes a data item and another reads or writes same data items, then order of execution is important.

Example

| T1 | T2 | T3 |
|---|---|---|
| Read(x) | | |
| | Write(x) | |
| | Commit | |
| Write(x) | | |
| Commit | | |
| | | Write(x) |
| | | Commit |

Schedule G

Precedence graph for above schedule is given below




Precedence Graph for Schedule G

Since, above precedence graph of the schedule G, with 3 transactions contains a cycle (of length 2; with two edges) through, the given schedule G is not conflict serializable.

9. ***Why do we need Concurrency control in non-serial transaction? Discuss time stamp based concurrency control with suitable non-serial transaction diagram example. (2+3=5)***

**Answer: What is Concurrency Control?**

Concurrency control is the procedure in DBMS for managing simultaneous operations without conflicting with each another. Concurrent access is quite easy if all users are just reading data.

Scanned by TapScanner

There is no way they can interfere with one another. Though for any practical database, would have a mix of reading and WRITE operations and hence the concurrency is a challenge. Concurrency control is used to address such conflicts which mostly occur with a multi-user system. It helps you to make sure that database transactions are performed concurrently without violating the data integrity of respective databases. Therefore, concurrency control is a most important element for the proper functioning of a system where two or multiple database transactions that require access to the same data, are executed simultaneously.

**Timestamp-based Protocols**

The timestamp-based algorithm uses a timestamp to serialize the execution of concurrent transactions. This protocol ensures that every conflicting read and write operations are executed in timestamp order. The protocol uses the System Time or Logical Count as a Timestamp. The older transaction is always given priority in this method. It uses system time to determine the time stamp of the transaction. This is the most commonly used concurrency protocol.

Lock-based protocols help you to manage the order between the conflicting transactions when they will execute. Timestamp-based protocols manage conflicts as soon as an operation is created.

**Example:** Suppose there are there transactions T1, T2, and T3.

T1 has entered the system at time 0010

T2 has entered the system at 0020

T3 has entered the system at 0030

Priority will be given to transaction T1, then transaction T2 and lastly Transaction T3.

**Advantages:**

Schedules are serializable just like 2PL protocols

No waiting for the transaction, which eliminates the possibility of deadlocks!

**Disadvantages:**

Starvation is possible if the same transaction is restarted and continually aborted

Rorito Lyk



10. *Explain any eight significant differences between a file processing system and a DBMS.* (5)

**Answer:** The difference between file processing system and database approach is as follow:

| File based system | Database system |
|---|---|
| Here the data and program are inter-dependent. | Here the data and program are independent of each other. |
| File-based system caused data redundancy. The data may be duplicated in different files | Database system control data redundancy. The data appeared only once in the system. |
| File based system caused data inconsistency. The data in different files may be different that cause data inconsistency. | In database system data always consistent. Because data appeared only once. |
| The data cannot be shared because data is distributed in different files. | In database data is easily shared because data is stored at one place. |
| In file based system data is widely spread. Due to this reason file based system provides poor security. | It provides many methods to maintain data security in the database. |
| File based system does not provide consistency constrains. | Database system provides a different consistency constrains to maintain data integrity in the system. |
| File based system is less complex system. | Database system is very complex system. |
| The cost of file processing system is less then database system. | The cost of database system is much more than a file processing system. |
| File based system takes much space in the system and memory is wasted in this approach. | Database approach store data more efficiently it takes less space in the system and memory is not wasted. |
| To generate different report to take a crucial decision is very difficult in file based system. | The report can be generated very easily in required format in database system. Because data in database is stored in an organized manner. And easily retrieve to generate different report. |
| File based system does not provide concurrency facility. | Database system provides concurrency facility. |
| File based system does not provide data atomicity functionality. | Database system provides data atomicity functionality. |
| The cost of file processing system is less than database system. | The cost of database system is more than file processing system. |
| It is difficult to maintain as it provides less controlling facility. | Database provides many facilities to maintain program. |
| If one application fail it does not affects other files in system. | If database fail it affects all application that dependent on database |
| Hardware cost is less than database system | Hardware cost is high in database than file system. |

Scanned by TapScanner

11. *Write down the steps to perform left outer join operation and right outer join operation of two relations R and S.* (5)

*Answer:* Left Outer Join(R ⟕ S)

All the tuples from the Left relation, R, are included in the resulting relation. If there are tuples in R without any matching tuple in the Right relation S, then the S-attributes of the resulting relation are made NULL.

| Course | |
|---|---|
| A | B |
| 100 | Database |
| 101 | Mechanics |
| 102 | Electronics |

| HOD | |
|---|---|
| A | B |
| 100 | Alex |
| 102 | Maya |
| 104 | Mira |

| Courses ⟕ HOD | | | |
|---|---|---|---|
| A | B | C | D |
| 100 | Database | 100 | Alex |
| 101 | Mechanics | — | — |
| 102 | Electronics | 102 | Maya |

**Right Outer Join:** (R ⟖ S)

All the tuples from the Right relation, S, are included in the resulting relation. If there are tuples in S without any matching tuple in R, then the R-attributes of resulting relation are made NULL.

| Courses ⟖ HOD | | | |
|---|---|---|---|
| A | B | C | D |
| 100 | Database | 100 | Alex |
| 102 | Electronics | 102 | Maya |
| — | — | 104 | Mira |

12. ***What is specialization? How it is differ from generalization? Explain aggregation in EER diagram with suitable example.*** (5)

*Answer:* Specialization

Specialization is opposite to Generalization. It is a top-down approach in which one higher level entity can be broken down into two lower level entities. In specialization, a higher level entity may not have any lower-level entity sets, it's possible. It

Rorito Lyk



maximizes the difference between the members of an entity by identifying the unique characteristic or attributes of each member. It defines one or more sub class for the super class and also forms the super class/subclass relationship.

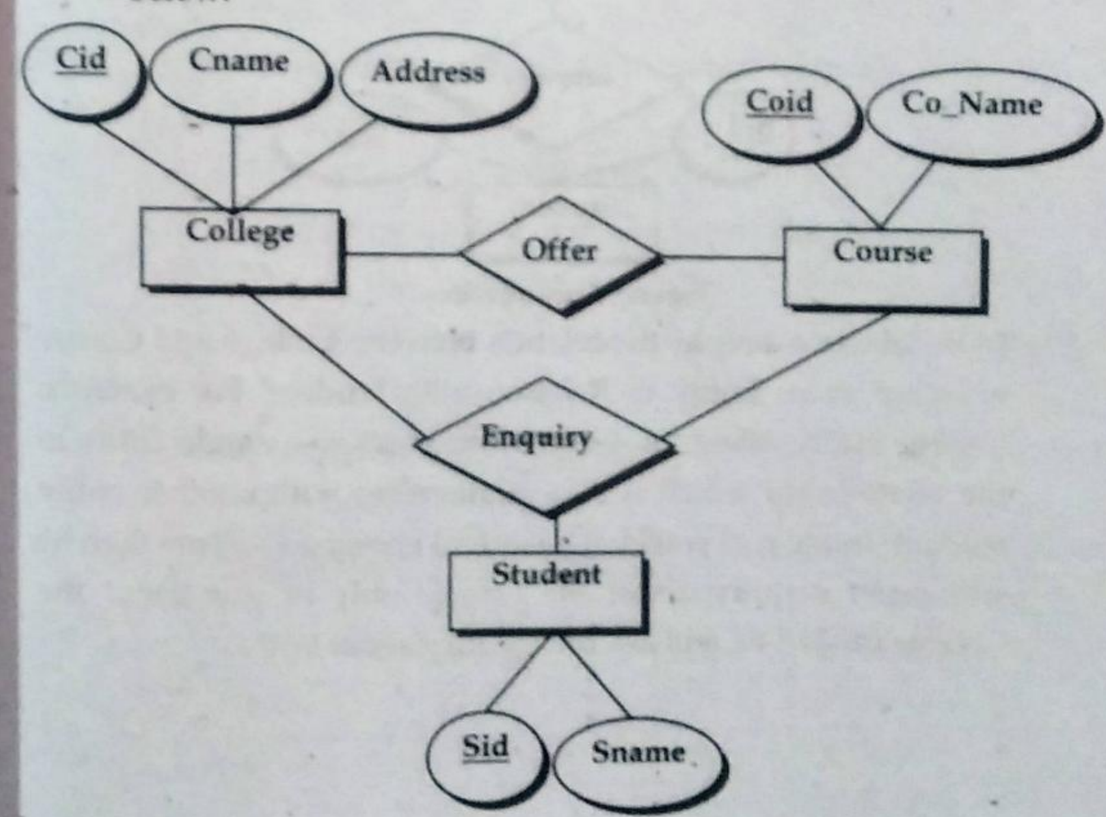
**Generalization**

Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalization, the higher level entity can also combine with other lower level entities to make further higher level entity.

It's more like Super class and Subclass system, but the only difference is the approach, which is bottom up. Hence, entities are combined to form a more generalized entity, in other words, sub-classes are combined to form a super-class.

**Aggregation**

The E-R model cannot express relationships among relationship sets and relationships between relationship sets and entity sets. Consider a DB with information about Colleges that offer courses and students enquiry colleges and courses offer by college. One alternative in this case is to define ternary relationship between {College, course, and Student} as in figure below.




Scanned by TapScanner

Main problem with above ER diagram is that it has redundant relationships. Every {Employee, Project} combination that appears in uses relationship set also appears in works relationship set. The solution is to use aggregation.

Aggregation is a process that represents a relationship between a whole object and its component parts. Using aggregation we can express relationship among relationships. Aggregation shows 'has-a' or 'is-part-of' relationship between entities where one represents the 'whole' and other 'part'. It abstracts a relationship between objects and viewing the relationship as an object. It is a process when two entity and relationship with these entities is treated as a higher level single entity set.

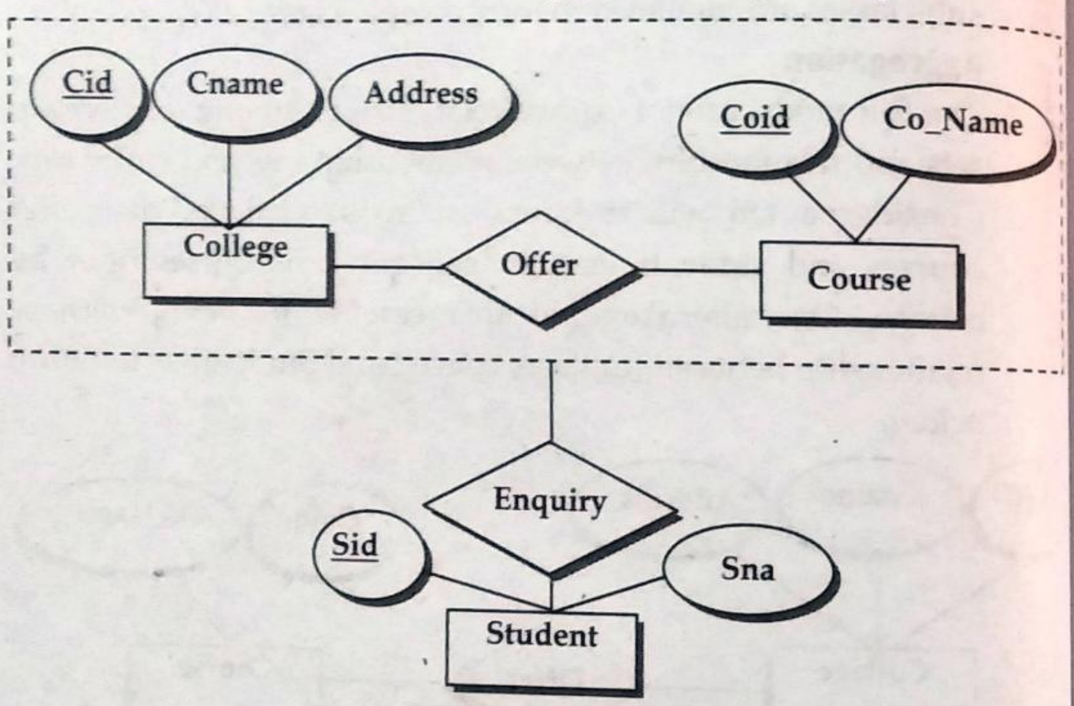



**Figure:** Aggregation

In the above example, the relation between College and Course is acting as an Entity in Relation with Student. **For example:** College entity offers the Course entity act as a single entity in the relationship which is in a relationship with another entity student. In the real world, if a student enquiry a college then he will never enquiry about the Course only or just about the College instead he will ask the enquiry about both.

Rorito Lyk



# Model Questions Sets For Practice

Course Title: Database Management System — Full Marks: 60
Course No: CSC260 — Pass Marks: 24
Semester: IV — Time: 3 Hrs

## MODEL SET 1

### Section A (Long questions)

**Attempt any two questions.** **(2 × 10 = 20)**

1. **Consider the following database and write SQL as given:** **(5 × 2 = 10)**

Teacher (Tid, Tname, Address, Age)
Student (Sid,Sname, Age, sex)
Takes (sid, course-id)
Course (course-id,course_name, text_book)
Teaches (Tid, Cousrse-id)
Taught-by{Sid, Tid}

a. Fine name, age and sex of all students who takes course "DBMS"
b. Find total number of students who are taught by the teacher "T01"
c. List all course names text books taught by teaher "T16"
d. Find average age of teachers for each course.
e. Insert the record of new teacher "T06" named "Dinesh" of age 27 into database who lives in "Balkhu" and takes course "DBMS"

2. What is normalization? Convert the following table to second normal form (2+8=10)

| Student ID | Student name | Prof ID | Prof Name | Grade |
|---|---|---|---|---|
| 1 | Ashwini | 101 | Dr. Shrestha | A |
| 2 | Rewati | 102 | Dr. Sharma | A |
| 3 | Raman | 103 | Dr. Bhusal | C |
| 4 | Krishna | 104 | Dr. Rimal | D |
| 5 | Hari | 105 | Dr. Khatri | B |

Scanned by TapScanner

3. Consider the following relational schema of library where primary keys are underlined. (5 x 2=10)

Employee (E-name, street, city)

Works (E-name, company-name, salary)

Company (company-name, city)

Manages (E-name, manager-name)

Write down RA for the following:

a. List out the names of all employees who work for ABC Company.

b. List out the names of all employees who do not work in any Company.

c. Count the total no. of employee work in Microsoft Company.

d. Insert a new record in Company table.

e. List out the name of employee and salary those are managed by James manager.

**Section B (Short questions)**

**Attempt any eight questions.** **(8 × 5 = 40)**

4. What is a weak entity set? How weak entity can be converted into strong entity? Give an example. (2 +3 = 5)

5. Make an ER diagram containing entities student, books, grade making your own assumptions about attributes and cardinality constraints. (5)

6. What are the advantages of specifying integrity constraints? Explain with an example. (2+3=5)

7. Differentiate between partial and transitive functional dependencies with example. (5)

8. Describe the serial and serializable schedule? Why serializable schedule is consider correct? (2+3=5)

9. How the granularity of data items does affect the performance of concurrency control? What factors affect selections of granularity size for data items? (2+3=5)

10. Differentiate between logical data independence and physical data independence. (5)

11. Compare the differences between cross join and natural join operations with suitable example. (5)

12. Why normalization is done? Explain insert, delete and update anomalies of database. (5)

Rorito Lyk



# MODEL SET 2

**Section A (Long questions)**

**Attempt any two questions.** **(2 × 10 = 20)**

1. **Consider the following database and write SQL as given:** **(5 × 2 = 10)**

Suppliers (sid: integer, sname: string, address: string)

Parts (pid: integer, pname: string, color: string)

Catalog (sid: integer, pid: integer, cost: real)

a. Find the pnames of parts for which there is some supplier.

b. Find the snames of suppliers who supply every part.

c. Find the snames of suppliers who supply every red part.

d. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.

e. Find the sids of suppliers who charge more for some part than the average cost of that Part (averaged over all the suppliers who supply that part).

2. What is functional dependency? Describe full, partial, transitive and trivial functional dependency with suitable example. (3+7=10)

3. In what sense does relational calculus differ from relational algebra, and in what sense they are similar? Describe any three additional operations of RA with suitable example. (4+6=10)

**Section B (Short questions)**

**Attempt any eight questions.** **(8 × 5 = 40)**

4. Draw a state diagram, and discuss the typical state that a transaction goes through during transaction. (5)

5. Draw an ER diagram for a database showing Hospital system. The Hospital maintains data about Affiliated Hospitals, type of Treatments facilities given at each hospital, and Patients (5)

6. How can define view in SQL? Explain the problems that may arise when one attempts to update a view. (2+3=5)

7. Define functional dependency. Describe the closure of a set of functional dependencies with an example. (2 + 3 = 5)

8. Which of the following schedule is (conflict) serializable? For each serializable schedule, determine the equivalent serial schedules. (2+2+1=5)

$r_1(x); r_3(x); w_1(x); r_2(x); w_3(x);$

$r_1(x); r_3(x); w_3(x); w_1(x); r_2(x);$

$r_3(x); r_2(x); r_1(x); w_3(x); w_1(x);$

Scanned by TapScanner

9. Discuss the problems of deadlock and starvation, and the different approaches to dealing with these problems. (2 + 3 = 5)
10. Differentiate between two-tier and three tier client/server architecture. (5)
11. What is constraint? How does SQL allow implementation of general integrity constraints? (5)
12. What is data abstraction? Discuss three levels of this abstraction. (5)

## MODEL SET 3

### Section A (Long questions)

**Attempt any two questions.** **(2 × 10 = 20)**

1. **Consider the following database and write SQL as given:** **(5 × 2 = 10)**

   Books (Bid,Btitle, Bauthor, Bpublisher, Bprice)
   Members ( Member_id, Name, Designation, Age)
   Reserve ( Member_id, Bid, Date)

   a. Create the tables using Books, Members and Reserve by specifying the Primary key, Not NULL, Foreign key Constraints DDL Statement in MySQL database
   b. Write SQL DML statement to insert any five tuples (Five records) in each relation(table)
   c. Find the Books of Database System title and price above 500.
   d. List the books published by 'KEC publication'
   e. Find the Name of the member who made reserve book in 12-10-2011
2. Define Boyce-Codd normal form. How does it differ from 3NF? Why is it considered a stronger form of 3NF? (2+2+6 = 10)
3. How can you mapping ER diagram with many to many relationship to relational model? Explain with suitable example. Also describe extended operations of RA with suitable example. (3+7 = 10)

### Section B (Short questions)

**Attempt any eight questions.** **(8 × 5 = 40)**

4. What is database management system? Discuss the advantages and disadvantages of using database management system over file system. (1+2+2=5)
5. Draw an ER diagram for database showing Bank. Each Bank can have multiple branches, and each branch can have multiple accounts and loans. (5)

Rorito Lyk



6. Define integrity constraint? Discuss domain constraint with suitable example. (2+3=5)
7. What are different update anomalies? Explain each in with suitable examples. (2+3=5)
8. What is serializable schedule? How can you test a schedule for conflict serializability? (2+3=5)
9. Why we need database recovery? Describe log based recovery with suitable practical example. (2+3=5)
10. Describe database system architecture with suitable block diagram. (5)
11. What is join operation? Differentiate between equijoin and natural join with suitable example (2+3=5)
12. Describe following terms: (2+3=5)
    a) The null value attribute and its uses.
    b) Recursive relationship type with suitable example.

## MODEL SET 4

### Section A (Long questions)

**Attempt any two questions.** **(2 × 10 = 20)**

1. **Consider the following database and write SQL as given:** **(5 × 2 = 10)**

   Suppliers (sid: integer, sname: string, address: string)
   Parts (pid: integer, pname: string, color: string)
   Catalog (sid: integer, pid: integer, cost: real)

   a. Find the pnames of parts for which there is some supplier.
   b. Find the snames of suppliers who supply every part.
   c. Find the snames of suppliers who supply every red part.
   d. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.
   e. Find the sids of suppliers who supply only red parts.
2. In what way do the generalized definitions of 2NF and 3NF extend the definitions beyond primary keys? Also compare 2 NF and 3NF. (6+4 = 10)
3. What is union compatibility? What are the relational algebra operators that require the relation on which they are applied be union compatible? Explain each of them with suitable practical example. (2+4+4 = 10)

Scanned by TapScanner

### Section B (Short questions)

**Attempt any eight questions.** (8 × 5 = 40)

4. The difference among a relationship instance, a relationship type, and relationship set. (5)
5. Consider a bus ticketing system that records information about the passenger, bus and route. Passenger is assigned to a bus travels to route. A bus contains many passengers and a passenger can be assigned into only one bus. Many buses travel in same route but a bus can travel in only one route. The attributes of passenger are pid (unique), gender and telephone (multi-valued). Similarly bus contains regno (unique) and color and route contains rid (unique), distance and rate (based on distance). Now draw the E-R diagram to represent this situation. (5)
6. Define primary key constraint? Discuss their properties with suitable example. (2+3=5)
7. Define non-loss decomposition with example. (2+3=5)
8. If deadlock is avoided by deadlock-avoidance schemes, is starvation still possible? Explain. (5)
9. What is checkpoint? Why is it needed? What are the actions which are taken by DBMS at a checkpoint? Explain with the help of an example. (1+1+1+2=5)
10. What is the use of ISA in EER diagram? Explain with suitable diagram. (5)
11. What is outer join operation in RA? Differentiate between outer join and natural join operations with suitable example. (1+4=5)
12. Describe following terms: (2+3=5)
    c) The multivalued attribute.
    d) Total vs. partial participation constraint in ER diagram

## MODEL SET 5

### Section A (Long questions)

**Attempt any two questions.** (2 × 10 = 20)

1. **Consider the following database and write SQL as given:** (5 × 2 = 10)

Student (snum: integer, sname: string, major: string, level: string, age: integer)
Class (cname: string, meets-at: time, room: string, fid: integer)
Enrolled (snum: integer, cname: string)
Faculty (fid: integer, fname: string, deptid: integer)



a. Find the names of all juniors (Level = JR) who are enrolled in a class taught by I. Teach.
b. Find the age of the oldest student who is either a History major or is enrolled in a course taught by I. Teach.
c. Find the names of all classes that either meet in room R128 or have five or more students enrolled.
d. Find the names of all students who are enrolled in two classes that meet at the same time.
e. Find the names of faculty members who teach in every room in which some class is taught.

2. You are given the below functional dependencies for relation R(A, B, C, D, E)
$F = \{AB \rightarrow C, AB \rightarrow D, D \rightarrow A, BC \rightarrow D, BC \rightarrow E\}$ (3+3+4 = 10)
    a. Is this relation is in BCNF? If not, show all dependencies that violate it.
    b. Is this relation in 3NF? If not, show all dependencies that violate it.
    c. Is the following dependency implied by the above set of dependencies? If so, show how using the Amstrong's Axioms : $ABC \rightarrow AE$

3. Give the following queries in the relational algebra using the relational schema (10 x 1=10)
Student (id, name)
EnrolledIn (id, code)
Subject (code, lecturer)
    a. What are the names of students enrolled in cs3020?
    b. Which subjects is Hector taking?
    c. Who teaches cs1500?
    d. Who teaches cs1500 or cs3020?
    e. Who teaches at least two different subjects?
    f. What are the names of students in cs1500 or cs3010?
    g. What are the names of students in both cs1500 and cs1200?
    h. What are the names of students in at least two different subjects?
    i. What are the codes of all the subjects taught?
    j. What are the names of all the students?

Scanned by TapScanner

## Section B (Short questions)

**Attempt any eight questions.** **(8 × 5 = 40)**

4. Differentiate between weak and strong entity set in ER diagram with suitable example. (5)
5. Assume we have the following application that models soccer teams, the games they play, and the players in each team. In the design, we want to capture the following: (5)
   - We have a set of teams, each team has an ID (unique identifier), name, main stadium, and to which city this team belongs.
   - Each team has many players, and each player belongs to one team. Each player has a number (unique identifier), name, DoB, start year, and shirt number that he uses.
   - Teams play matches, in each match there is a host team and a guest team. The match takes place in the stadium of the host team.
   - For each match we need to keep track of the following:
     - ✓ The date on which the game is played
     - ✓ The final result of the match
     - ✓ The players participated in the match. For each player, how many goals he scored, whether or not he took yellow card, and whether or not he took red card.
     - ✓ During the match, one player may substitute another player. We want to capture this substitution and the time at which it took place.
   - Each match has exactly three referees. For each referee we have an ID (unique identifier), name, DOB, years of experience. One referee is the main referee and the other two are assistant referee.

   Design an ER diagram to capture the above requirements. State any assumptions you have that affect your design (use the back of the page if needed). Make sure cardinalities and primary keys are clear.
6. What is the use of trigger in database? Explain use of trigger in database with suitable practical example. (2+3=5)
7. What is key in database? List out possible keys used in database and explain them with example. (2+3=5)
8. What is transaction? Describe ACID properties of transaction with suitable example. (1+4=5)
9. Explain the validation techniques and snapshot isolation concurrency control in detail. (5)



10. What is the use aggregation in EER diagram? Explain with suitable diagram. (5)
11. What is theta join operation in RA? Differentiate between Equi join and theta join operations with suitable example. (2+3=5)
12. Describe following terms: (2+3=5)
    e) Supper class sub-class relationship.
    f) Domain constraint

## MODEL SET 6

### Section A (Long questions)

**Attempt any two questions.** **(2 × 10 = 20)**

1. **Consider the following database and write SQL as given:** **(5 × 2 = 10)**

   Sailors (sid: integer, sname: string, rating: integer, age: real)
   Boats (bid: integer, bname: string, color: string)
   Reserves (sid: integer, bid: integer, day: date)
   a. Find only the Sailors with the name equal to "Ajaya" AND the age equal to 33.
   b. Find the sids of sailors who have reserved a red boat.
   c. Selects all Sailors with age between 20 and 40
   d. Find the names of sailors who have reserved a red or a green boat.
   e. Find average age of all sailors with a rating of 9.
2. What is functional dependency? How BCNF differ from 3-NF? Justify with suitable example.
3. Consider the following insurance database, where primary keys are underlined:

   Student (StuName, StuId, Class, Major)
   Course (CourseName, CourseNumber, CreditHours, Department)
   Section (SectionId, CourseNumber, Semester, Year, Instructor-Name)
   GradeReport (StuId, SectionId, Grade)
   Prerequisite (CourseNumber, PrerequisiteNumber)

   Construct the following RA expressions for this relational database
   a. Display name and Id of those students whose major is finance and study in 4th semester.
   b. Display the number of courses taught by 'Susant'

Scanned by TapScanner

c. Change credit hours of course which course number equal to 19 to 44.

d. Insert a new student {'Raju', 10, 12, 'DBMS'} in student relation.

e. Display student name, class and instructor name who taught "DBMS"

**Section B (Short questions)**

**Attempt any eight questions.** **(8 × 5 = 40)**

4. Explain different types of cardinality constraints used in ER diagram with practical example. (5)
5. Make E-R diagram on the basis of following scenario. This is system where information about drivers, bus, tickets and customers are stored. Drivers drive bus. Customers travel from one place to another by the bus. Driver checks the ticket of each customer for validity. Customers have name, contact number, and age. Each bus has unique bus number, its capacity, and type. Tickets have source, destination, price and date. Each customer must have ticket to travel by the bus.
6. What is UNIQUE and NOT NULL constraint in SQL? How these constraints are differing from primary key constraint? Explain with suitable example. (1+2+2=5)
7. How flat file system differ from database system? Explain few advantages of flat file system over database system. (2+3=5)
8. What is deadlock? Discuss deadlock prevention techniques with example. (1+4=5)
9. What do you mean by concurrency control? Why it is used? Explain time based concurrency control with suitable example. (5)
10. How ER with multivalued attribute is reduced into relational model? Explain with suitable example. (5)
11. What is DDL statement in SQL? How it is differ from DML statement? Explain with suitable example. (1+2+2=5)
12. Describe following terms: (1.5+1.5+2=5)
    a. Schedule
    b. Assertion
    c. Trigger

Rorito Ly



## MODEL SET 7

**Section A (Long questions)**

**Attempt any two questions.** **(2 × 10 = 20)**

1. **Consider the following database and write SQL as given:** **(5 × 2 = 10)**

   Employee (E-name, street, city)
   Works (E-name, company-name, salary)
   Company (company-name, city)
   Manages (E-name, manager-name)

   a. List out the names of all employees who work for ABC Company.
   b. List out the names of all employees who do not work in any Company.
   c. Count the total no. of employee work in Microsoft Company.
   d. Insert a new record in Company table.
   e. List out the name of employee and salary those are managed by James manager.
2. What problems will arise if partial dependency on a relation is not removed? Explain with example. Also compare partial dependency with fully functional dependency.
3. List out database modification operations in RA. Explain each of them with suitable RA expressions and their output pattern.

**Section B (Short questions)**

**Attempt any eight questions.** **(8 × 5 = 40)**

4. What is main difference between WHERE and HAVING clause? Explain with suitable practical example. (5)
5. Suppose you are given the following requirements for a simple database for the National Hockey League (NHL):
   - the NHL has many teams,
   - each team has a name, a city, a coach, a captain, and a set of players,
   - each player belongs to only one team,
   - each player has a name, a position (such as left wing or goalie), a skill level, and a set of injury records,
   - a team captain is also a player,
   - A game is played between two teams (referred to as host_team and guest_team) and has a date (such as May 11th, 1999) and a score (such as 4 to 2).

   Construct a clean and concise ER diagram for the NHL database.

Scanned by TapScanner

6. A union, intersection or difference can only be performed between two relations if they are type compatible. What is meant by type compatibility? Give an example of two type compatible and two non-type compatible relations? (3+2=5)
7. Illustrate the differences between hierarchical and network data models. Explain why relational model is preferred over the two. (2+3=5)
8. Define serializability. Differentiate conflict and view serializability. (1+4=5)
9. Identify whether the following schedule is deadlock free or not with the help of wait-for graph. S=R1(x), R2(y), W1(x), R2(x), R3(z), W3(z), R1(y), R3(x), W1(y). (5)
10. Compare the shadow-paging recovery scheme with the log-based recovery schemes in terms of ease of implementation and overhead cost. (5)
11. What is the use of LIKE operator in SQL? Explain the use of LIKE and IN operators in SQL with suitable practical example. (1+2+2=5)
12. Describe following terms: (1.5+1.5+2=5)
    a. Tuple relational calculus
    b. Concurrent schedule
    c. Referential integrity constraint

## MODEL SET 8

### Section A (Long questions)

**Attempt any two questions.** **(2 × 10 = 20)**

**1. Consider the following database and write SQL as given:** **(2 × 5 = 10)**

a. Write SQL ALTER command to add following attributes to table named "Employee"

| Name of attribute | Data type | Constraint |
|---|---|---|
| Epost | Text | Either manager or Accountant |
| Esalary | Integer | >=20000 and <=45000 |

b. Write SQL statement to create named student with attributes name, age and gender. Data type and constraints on attributes are as follows:

| Attribute | Data type | Constraint |
|---|---|---|
| Name | Text (30) | Not null |
| Age | Int | >10 |
| Gender | Char (1) | 'm' or 'f' |



2. What are different update anomalies? Explain each in with suitable examples.
3. Consider the following supplier database, where primary keys are underlined.
   Supplier (supplier-id, supplier-name, city)
   Supplies (supplier-id, part-id, quantity)
   Parts (part-id, part-name, color, weight)
   Construct the following relational algebra queries for this relational database
   a. Find the name of all suppliers located in city "Kathmandu".
   b. Find the name of all parts supplied "ABC Company".
   c. Find the name of all parts that are supplied in quantity greater than 300.
   d. Find the number of parts supplied by "ABC Company".
   e. Find the name of all suppliers who supply more than 30 different parts.

### Section B (Short questions)

**Attempt any eight questions.** **(8 × 5 = 40)**

4. Differentiate between database schema and a database state. (5)
5. A university registrar's office maintains data about the following entities:
   a. Courses, including number, title, credits, syllabus, and prerequisites;
   b. Course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom;
   c. Students, including student-id, name, and program;
   d. Instructors, including identi-cation number, name, department, and title.

   Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled. Construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraints. (5)
6. What is a view in SQL and how it is defined? Explain how views are typically implemented. (3+2=5)
7. Illustrate the differences between object oriented and object relational data models. Explain why relational model is preferred over the two. (2+3=5)
8. Discuss the problems of deadlock and starvation, and the different approaches to dealing with these problems. (5)

Scanned by TapScanner

9. Draw a state diagram, and discuss the typical state that a transaction goes through during transaction. (5)
10. How does the granularity of data items affect the performance of concurrency control? What factors affect selection of granularity size for data items? (3+2=5)
11. The difference among a relationship instance, a relationship type, and a relationship set. (5)
12. Describe following terms: (1.5+1.5+2=5)
    a. Domain relational calculus
    b. Serial and non-serial schedule
    c. Strong vs. weak entity set

## MODEL SET 9

### Section A (Long questions)

**Attempt any two questions.** **(2 × 10 = 20)**

1. **Consider the following database and write SQL as given:** **(5 × 2 = 10)**

   Professor (ssn, profname, status)
   Course (crscode, crsname, credits)
   Taught (crscode, ssn, semester)
   a. Display the name of all professors.
   b. Display the name of those professors whose status is unknown.
   c. List all the course names of 3 credit hours.
   d. Update the statues of professor "Suwarna" with "Full"
   e. Delete the professor "Sasidhar" from the professor relation
2. Distinguish 1NF, 2NF and 3NF in database design. How can the weaknesses prevalent in non 1NF relations be remedied? (5+5=10)
3. Consider the following supplier database, where primary keys are underlined.
   Supplier (supplier-id, supplier-name, city)
   Supplies (supplier-id, part-id, quantity)
   Parts (part-id, part-name, color, weight)
   Construct the following relational algebra queries for this relational database
   a. Find the name of all suppliers located in city "Kathmandu" and supply quantity 50.
   b. Find the name of all parts supplied "TATA Company".



   c. Insert any two records into supplier table.
   d. Find the number of parts supplied by "TATA Company".
   e. Find the name and city of all suppliers who supply more than 30 different parts.

### Section B (Short questions)

**Attempt any eight questions.** **(8 × 5 = 40)**

4. Define FD closure and attribute closure. Find any three members of F+ in the set of functional dependencies given below. (5)
   R = (A, B, C, G, H, I)
   F = {A → B, A → C, CG → H, CG → I, B → H}
5. Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted. (5)
6. What is database schema and database instance? Explain responsibilities of DBA in brief. (3+2=5)
7. What is data mode? Explain in briefly different types of data models with suitable example. (2+3=5)
8. Why database recovery is needed? Describe No-Undo/Redo and Undo/No-Redo database recovery techniques. (5)
9. What is lock? Describe two phase locking algorithm with suitable example. (5)
10. Explain the purpose of the checkpoint mechanism. How often should checkpoints be performed? How does the frequency of checkpoints affect: (5)
    i) System performance where no failure occurs
    ii) The time it takes to recover from a system crash
    iii) The time it takes to recover from a disk crash
11. What is the use of LIKE operator in SQL? Compare WHERE and HAVING clause in SQL. (5)
12. Describe following terms: (1.5+2+1.5=5)
    a. Concurrency control
    b. Log based recovery
    c. Composite key

❑❑❑

Scanned by TapScanner

# Theory of Computation

Course Title: Theory of Computation
Course No: CSC257
Nature of the Course: Theory + Lab
Semester: IV

Full Marks: 60 + 20 + 2
Pass Marks: 24 + 8 +
Credit Hrs: 3

**Course Objectives**

This course presents a study of Finite State Machines and their languages. It covers the details of finite state automata, regular expressions, context free grammars. More, the course includes design of the Push-down automata and Turing Machines. The course also includes basics of undecidabilty and intractability.

**Course Description**

The main objective of the course is to introduce concepts of the models of computation and formal language approach to computation. The general objectives of this course are to, introduce concepts in automata theory and theory of computation, design different finite state machines and grammars and recognizers for different formal languages, identify different formal language classes and their relationships, determine the decidability and intractability of computational problems.

**Course Details**

**Unit I: Basic Foundations** **(3 Hrs.)**

1.1. Review of Set Theory, Logic, Functions, Proofs

1.2. Automata, Computability and Complexity: Complexity Theory, Computability Theory, Automata Theory

1.3. Basic concepts of Automata Theory: Alphabets, Power of Alphabet, Kleen Closure Alphabet, Positive Closure of Alphabet, Strings, Empty String, Substring of a string, Concatenation of strings, Languages, Empty Language

**Unit II: Introduction to Finite Automata** **(8 Hrs.)**

2.1 Introduction to Finite Automata, Introduction of Finite State Machine

2.2 Deterministic Finite Automata (DFA), Notations for DFA, Language of DFA, Extended Transition Function of DFA Non-Deterministic Finite Automaton (NFA), Notations for NFA, Language of NFA, Extended Transition

2.3 Equivalence of DFA and NFA, Subset-Construction

2.4 Method for reduction of NFA to DFA, Theorems for equivalence of Language accepted by DFA and NFA

2.5 Finite Automaton with Epsilon Transition (ε - NFA), Notations for ε - NFA, Epsilon Closure of a State, Extended Transition Function of ε - NFA, Removing Epsilon Transition using the concept of Epsilon Closure, Equivalence of NFA and ε -NFA, Equivalence of DFA and ε - NFA



2.6 Finite State Machines with output: Moore machine and Mealy Machines

**Unit III: Regular Expressions** **(6 Hrs.)**

3.1 Regular Expressions, Regular Operators, Regular Languages and their applications, Algebraic Rules for Regular Expressions

3.2 Equivalence of Regular Expression and Finite Automata, Reduction of Regular Expression to ε - NFA, Conversion of DFA to Regular Expression

3.3 Properties of Regular Languages, Pumping Lemma, Application of Pumping Lemma, Closure Properties of Regular Languages over (Union, Intersection, Complement) Minimization of Finite State Machines: Table Filling Algorithm

**Unit IV: Context Free Grammar** **(9 Hrs.)**

4.1 Introduction to Context Free Grammar (CFG), Components of CFG, Use of CFG, Context Free Language (CFL)

4.2 Types of derivations: Bottomup and Topdown approach, Leftmost and Rightmost, Language of a grammar

4.3 Parse tree and its construction, Ambiguous grammar, Use of parse tree to show ambiguity in grammar

4.4 Regular Grammars: Right Linear and Left Linear, Equivalence of regular grammar and finite automata

4.5 Simplification of CFG: Removal of Useless symbols, Nullable Symbols, and Unit Productions, Chomsky Normal Form (CNF), Greibach Normal Form (GNF), Backus- Naur Form (BNF)

4.6 Context Sensitive Grammar, Chomsky Hierarchy Pumping Lemma for CFL, Application of Pumping Lemma, Closure Properties of CFL

**Unit V: Push Down Automata** **(7 Hrs.)**

5.1 Introduction to Push Down Automata (PDA), Representation of PDA, Operations of
PDA; Move of a PDA, Instantaneous Description for PDA

5.2 Deterministic PDA, Non Deterministic PDA, Acceptance of strings by PDA, Language of PDA

5.3 Construction of PDA by Final State , Construction of PDA by Empty Stack,

5.4 Conversion of PDA by Final State to PDA accepting by Empty Stack and vice-versa, Conversion of CFG to PDA, Conversion of PDA to CFG

**Unit VI: Turing Machines (10 Hrs.)**

6.1 Introduction to Turing Machines (TM), Notations of Turing Machine, Language of a Turing Machine, Instantaneous Description for Turing Machine, Acceptance of a string by a Turing Machines

Scanned by TapScanner

6.2 Turing Machine as a Language Recognizer, Turing Machine as a Computing Function, Turing Machine with Storage in its State, Turing Machine as a enumerator of stings of a language, Turing Machine as Subroutine

6.3 Turing Machine with Multiple Tracks, Turing Machine with Multiple Tapes, Equivalence of Multitape-TM and Multitrack-TM, Non-Deterministic Turing Machines, Restricted Turing Machines: With Semi-infinite Tape, Multistack Machines, Counter Machines

6.4 Curch Turing Thesis, Universal Turing Machine, Turing Machine and Computers, Encoding of Turing Machine, Enumerating Binary Strings, Codes of Turing Machine, Universal Turing Machine for encoding of Turing Machine

**Unit VII: Undecidability and Intractability** **(5 Hrs.)**

7.1 Computational Complexity, Time and Space complexity of A Turing Machine, Intractability

7.2 Complexity Classes, Problem and its types: Absract, Decision, Optimization

7.3 Reducibility, Turing Reducible, Circuit Satisfiability, Cook's Theorem,

7.4 Undecidability, Undecidable Problems: Post's Correspondence Problem, Halting Problem and its proof, Undecidable Problem about Turing Machines

**Laboratory Works:**

The laboratory work consists of design and implementation of finite state machines like DFA,

NFA, PDA, and Turing Machine. Students are highly recommended to construct Tokenizers/ Lexers over/for some language. Students are advised to use regex and Perl (for using regular expressions), or any other higher level language for the laboratory works.

**Text Books:**

1. **John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Pearson - Addison-Wesley.**

**Reference Books:**

1. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, 2nd Edition, Prentice Hall.
2. Michael Sipser, Introduction to the Theory of Computation, 3rd Edition, Thomson Course Technology
3. Efim Kinber, Carl Smith, Theory of Computing: A Gentle introduction, Prentice- Hall.
4. John Martin, Introduction to Languages and the Theory of Computation, 3rd Edition, Tata McGraw Hill.
5. Kenneth H. Rosen, Discrete Mathematics and its Applications to Computers Science, WCB/Mc-Graw Hill.



## TRIBHUVAN UNIVERSITY

**Institution of Science and Technology**

Course Title: Theory of Computation — Full Marks: 60
Course No: CSC257 — Pass Marks: 24
Level: B. Sc CSIT Second Year/ Fourth Semester — Time: 3 Hrs

## T.U. MODEL QUESTION-ANSWERS SET I

**Section A**

**Long Answer Questions**

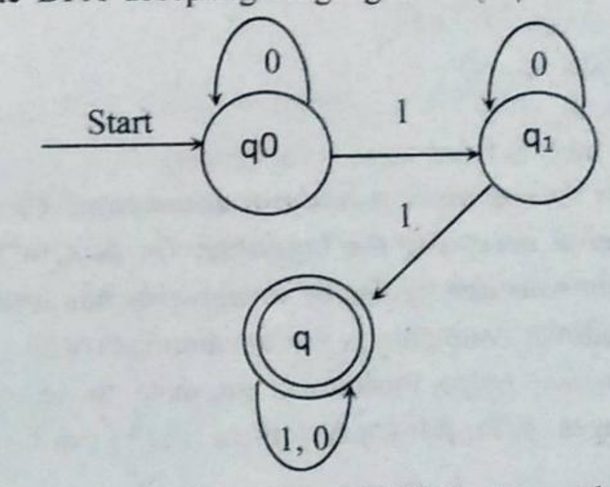
**Attempt any two questions.** **[2*10=20]**

1. ***Define the extended transition function of DFA. Draw a DFA accepting language L= {$1^n$\|n=2, 3, 4.....}. Show acceptance of strings 1110011 and 1110 using extended transition function.*** *[2+4+4]*

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

i.e. (q, w) = p

**Example**: The DFA accepting language L= {$1^n$\|n=2, 3, 4.....} is given below;




**Showing acceptance of string 1110011 by using extended transaction function as,**

$(q_0, 1110011)$

$= \delta(\hat{\delta}(q_0, 111001), 1)$

$= \delta(\delta(\hat{\delta}(q_0, 1110), 0), 1)$

$= \delta(\delta(\delta(\hat{\delta}(q_0, 1110), 0), 1), 1)$

Scanned by TapScanner

$= \delta(\delta(\delta(\delta(\hat{\delta}(q_0, 111), 0), 0), 1), 1)$

$= \delta(\delta(\delta(\delta(\delta(\hat{\delta}(q_0, 11), 1), 0), 0), 1), 1)$

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

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

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

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

$= \delta(\delta(\delta(\delta(\delta(q_2, 1), 0), 0), 1), 1)$

$= \delta(\delta(\delta(\delta(q_2, 0), 0), 1), 1)$

$= \delta(\delta(\delta(q_2, 0), 1), 1)$

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

$= \delta(q_2, 1)$

$= q_2$ which is final state so accepted

**Showing acceptance of string 1110 by using extended transaction function as,**

$\hat{\delta}(q_0, 1110)$

$= \delta(\hat{\delta}(q_0, 111), 0)$

$= \delta(\delta(\hat{\delta}(q_0, 11), 1), 0)$

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

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

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

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

$= \delta(\delta(q_2, 1), , 0)$

$= \delta(q_2, 0)$

$= q_2$ which is final state so accepted

2. ***What is deterministic pushdown automaton? Configure a pushdown automaton accepting the language, L= {w C $w^R$|w Є (a, b)*}. Show instantaneous description of strings abbCbba and baCba.*** *[2+4+4]*

**Answer:** Pushdown Automata is a finite automata with extra memory called stack which helps Pushdown automata to recognize Context Free Languages. A Pushdown Automata (PDA) can be formally defined as follows:

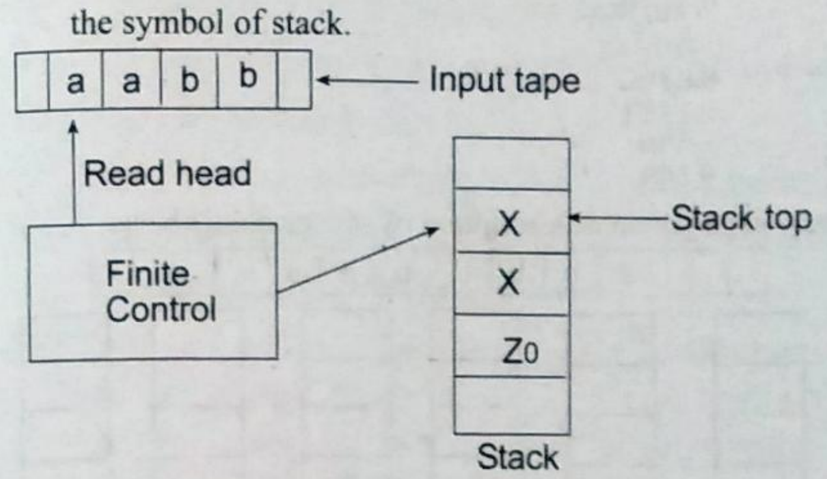
P = (Q, ∑, Γ, δ, q0, Z0, F) where,

- Q is the set of states
- ∑ is the set of input symbols
- Γ is the set of pushdown symbols (which can be pushed and popped from stack)
- q0 is the initial state

Rorito Lyk



- Z0 is the initial pushdown symbol (which is initially present in stack)
- F is the set of final states
- δ is a transition function which maps Q x {∑ ∪ Є} x Γ into Q x Γ*. In a given state, PDA will read input symbol and stack symbol (top of the stack) and move to a new state and change the symbol of stack.




Pushdown automaton accepting the language, L= {w C $w^R$ | w Є (a, b)*}

By analysis of language it must satisfy following strings

{abcba, bacab, ababcbaba, .....}

Let push down automata be, P = (Q, ∑, Γ, δ, $q_0$, $Z_0$, F)

Where, Q= {$q_0$, $q_1$, qf}

∑ = {a, b}

Γ = {a, b, $z_0$}

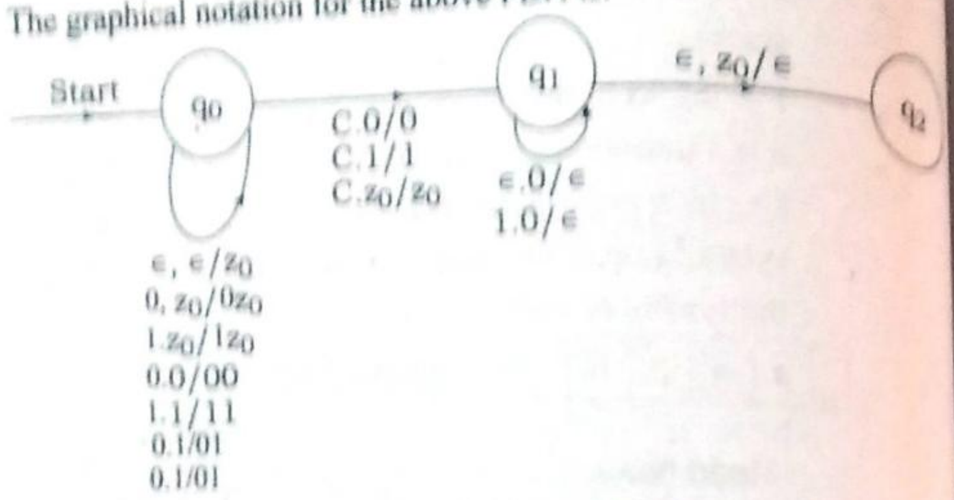
qf = Final state

And transaction function δ can be defined as follows,

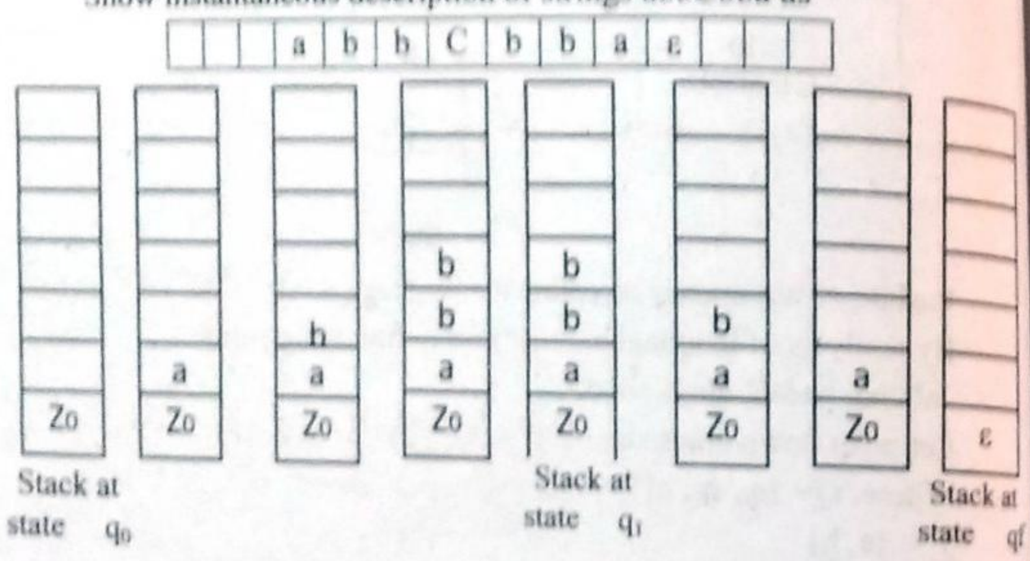
1. δ( $q_0$, ε , ε) = ( q0, Z0)
2. δ( $q_0$, a, $Z_0$) = ( q0, aZ0)
3. δ( q0, b, Z0) = ( q0, bZ0)
4. δ( q0, a, a) = ( q0, aa)
5. δ( q0, b, b) = ( q0, bb)
6. δ( q0, a, b) = ( q0, ab)
7. δ( q0, b, a) = ( q0, ba)
8. δ( q0, c, a) = ( q1, a) //Change the state
9. δ( q0, c, b) = ( q1, b) // change the state
10. δ( q0, c, z0) = ( q1, z0) //change the state
11. δ( q1, a, a) = ( q1, ε)
12. δ( q1, b, b) = ( q1, ε)
13. δ( q1, ε, z0) = ( qf, ε)

Scanned by TapScanner

The graphical notation for the above PDA is:




Show instantaneous description of strings abbCbba as




$(q_0, abbcbba, z_0) = (q_0, bbcbba, az_0)$
$= (q_0, bcbba, baz_0)$
$= (q_0, cbba, bbaz_0)$
$= (q_1, bba, bbaz_0)$
$= (q_1, ba, baz_0)$
$= (q_1, a, az_0)$
$= (q_1, \varepsilon, z_0)$
$= (qf, \varepsilon, \varepsilon)$ accepted

Show instantaneous description of strings baCba

$(q_0, bacba, z_0) = (q_0, acba, bz_0)$
$= (q_0, cba, abz_0)$
$= (q_1, ba, abz_0)$
$= (q_1, ba, abz_0)$ not accepted

3. ***How a Turing Machine works? Construct a Turing Machine accepting the language, L= {(ⁿ)ⁿ}. Also show the transition diagram of the machine. Illustrate whether a string (( )) is accepted by the Turing Machine or not.*** *[2+6+2]*

**Answer:** Turing Machine was invented by Alan Turing in 1936 and it is used to accept Recursive Enumerable Languages (generated by Type-0 Grammar). A Turing machine consists of a tape of infinite length on

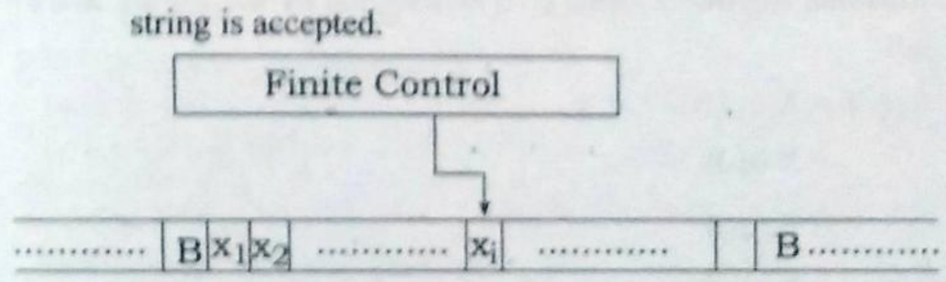


which read and writes operation can be performed. The tape consists of infinite cells on which each cell either contains input symbol or a special symbol called blank. It also consists of a head pointer which points to cell currently being read and it can move in both directions. A TM is expressed as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q0, B, F)$ where:

- $Q$ = the finite set of states of the finite control
- $\Sigma$ = the finite set of input symbols
- $\Gamma$ = the complete set of tape symbols $\Sigma$ is always subset of $\Gamma$.
- $q_0$ = the start state; $q0 \varepsilon Q$
- $B$ = the blank symbol; $B \varepsilon \Gamma$ but B does not belong to $\Sigma$.
- $F$ = the set of final or accepting states; F is subset of Q
- $\delta$ = the transition function defined by

$Q \times \Gamma \rightarrow Q \times \Gamma \times (R, L, S)$; where R, L, S is the direction of movement of head left, or right or stationary. I.e. $d(q, x) = d(p, Y, D)$; which means T M in state q and current tape symbol x, moves to next state P, replacing tape symbol x with Y and move the head either direction or remains at same cell of input tape.

- $q_0$ is the initial state
- F is the set of final states. If any state of F is reached, input string is accepted.




**Turing Machine accepting the language, L= {(ⁿ)ⁿ}**

We are going to design a Turing machine which accepts a set of strings in which every string starts with '('followed by any number of '('. Every string ends in equal number of ')' as '('. No '('is encountered after first ')' is read. It will not accepts the strings like {ε, (, ((, (()), ()), ), )), (((} and so on }

Let required Turing machine is:

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

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

$\Sigma = \{(,)\}$

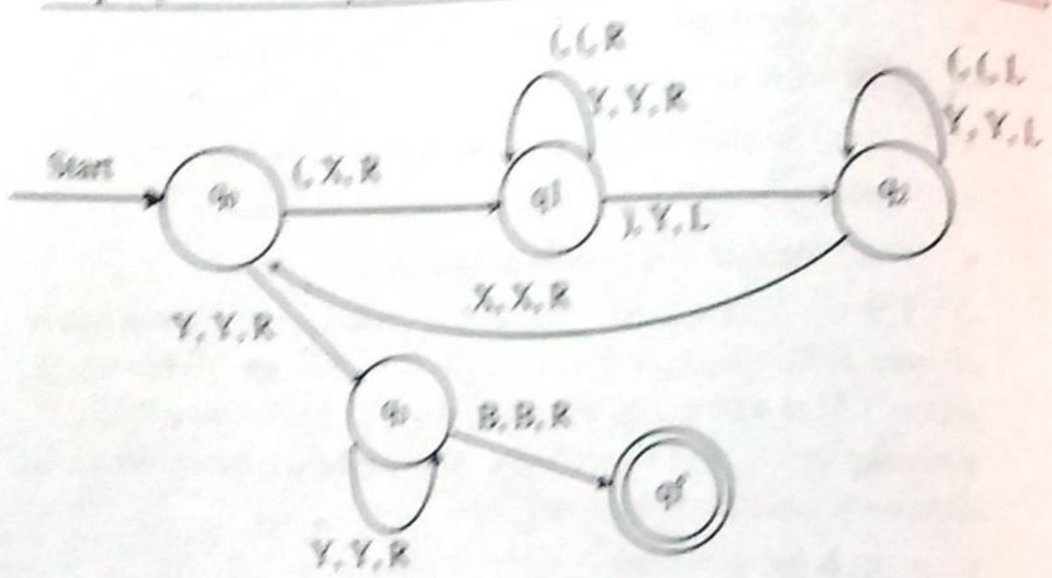
$\Gamma = \{(, ), X, Y\}$

A

Scanned by TapScanner

and δ can be defined as

| | ( | ) | X | Y | B |
|---|---|---|---|---|---|
| $q_0$ | $(q_1, X, R)$ | | | $(q_0, Y, R)$ | |
| $q_1$ | $(q_1, (, R)$ | $(q_2, Y, L)$ | | $(q_1, Y, R)$ | |
| $q_2$ | $(q_2, (, L)$ | | $(q_0, X, R)$ | $(q_2, Y, L)$ | |
| $q_3$ | | | | $(q_3, Y, R)$ | $(q^f, B, R)$ |
| $q^f$ | | | | | Accept |




Illustrate whether a string (( )) is accepted by the Turing Machine or not:

$q_0$ (( )) = X $q_1$ ())

= X ($q_1$))

= X ($q_1$))

= X $q_2$(Y)

= $q_2$ X (Y)

= X $q_0$ (Y)

= X X $q_1$ Y)

= X X Y $q_1$)

= X X $q_2$YY

= X $q_2$ X YY

= X X $q_0$ YY

= X X Y $q_3$Y

= X X Y Y $q_3$ B

= qf Accepted



## Section B

**Short Answer Questions**

**Attempt any eight questions.**

4. **When a grammar is said to be in CNF? Convert following grammar to CNF;** [B+5=40]

**S→ 1A | 0B | ε** [1+4]

**A→ 1AA | 0S | 0**

**B→ 0BB | 1 |A**

**C→CA | CS**

**Answer:** A context free grammar G = (V, T, P, S) is said to be in Chomsky's Normal Form (CNF) if every production in G are in one of the two forms;

A→ BC and

A→ a where {A, B, C} ε V and a ε T

Thus a grammar in CNF is one which should not have;

- ε-production
- Unit production
- Useless symbols.

**Second part:**

Convert following grammar to CNF

S→ 1A | 0B | ε

A→ 1AA | 0S | 0

B→ 0BB | 1 |A

C→CA | CS

At first remove ε-production as,

S→ 1A | 0B

A→ 1AA | 0S | 0

B→ 0BB | 1 |A

C→CA | CS |C

Now remove unit production as,

S→ 1A | 0B

A→ 1AA | 0S | 0

B→ 0BB | 1 |1AA |0S | 0

C→CA | CS

Also remove useless symbols as,

Since C is unreachable from starting symbol S so last production is useless,

S→ 1A | 0B

A→ 1AA | 0S | 0

B→ 0BB | 1 |1AA |0S | 0

Now convert to CNF as

S→ $A_1$A | $B_1$B

A→ $A_1$AA | $B_1$S | 0

B→ $B_1$BB | 1 |$A_1$AA | 0

$A_1$→ 1

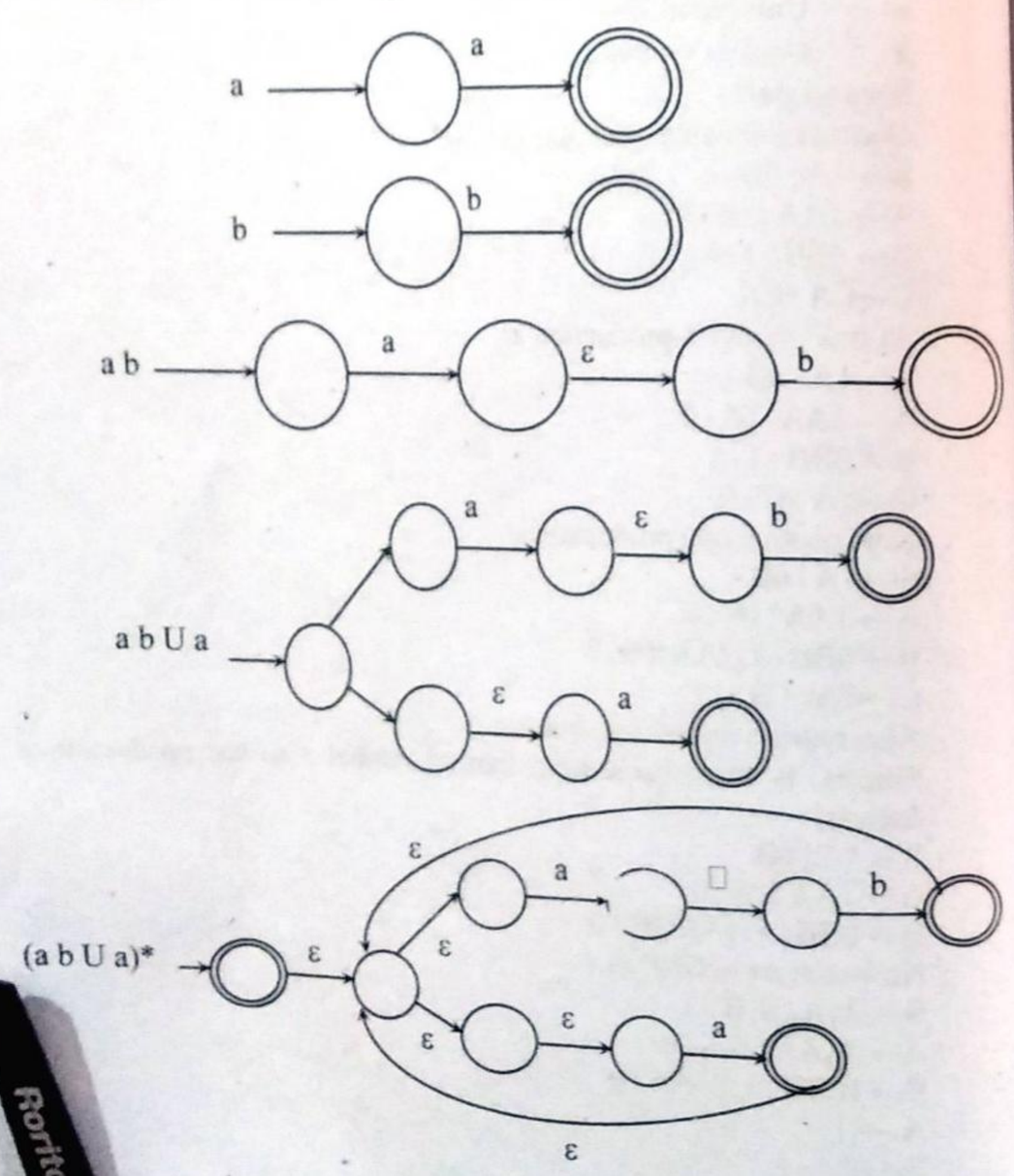
$B_1$→0

Scanned by TapScanner

5. ***Define epsilon NFA. Configure equivalent epsilon NFA for the regular expression (ab U a)*.*** *[1+4]*

**Answer:** Nondeterministic finite automaton with ε-moves (NFA-ε) is a further generalization to NFA. This automaton replaces the transition function with the one that allows the empty string ε as a possible input. The transitions without consuming an input symbol are called ε-transitions. In the state diagrams, they are usually labeled with the Greek letter ε. ε-transitions provide a convenient way of modeling the systems whose current states are not precisely known: i.e., if we are modeling a system and it is not clear whether the current state (after processing some input string) should be q or q', then we can add an ε-transition between these two states, thus putting the automaton in both states simultaneously.

**Epsilon NFA for the regular expression (ab U a)***






6. ***Differentiate Kleene Closure from Positive Closure. For ∑ = {0, 1}, compute ∑* and ∑2.***

**Answer:** The Kleene closure denoted by ∑*, is a unary operator on a set of symbols or strings, ∑, that gives the infinite set of all possible strings of all possible lengths over ∑ including ε.

Representation – ∑* = ∑0 ∪ ∑1 ∪ ∑2 ∪..... where ∑ p is the set of all possible strings of length p.

**Example** – If ∑ = {a, b}, ∑* = {ε, a, b, aa, ab, ba, bb…}

The positive closure denoted by ∑+ is the infinite set of all possible strings of all possible lengths over ∑ excluding ε.

Representation – ∑+ = ∑1 ∪ ∑2 ∪ ∑3 ∪.....

∑+ = ∑* – { ε }

**Example:** If ∑ = {a, b}, ∑+ = {a, b, aa, ab, ba, bb...}

**For ∑ = {0, 1}, compute ∑* and ∑2**

∑* = {ε, 0, 1, 00, 11, 10, 01, 000, 111, 101, 010, 1010…}

∑2 = {00, 11, 10, 01}

7. ***Write the regular expression over {0, 1} for strings*** *[2.5+2.5]*
   - ***Not ending with 0.***
   - ***Of length at least 3 that ends with 00.***

**Answer:** Not ending with 0.

RE: (0+1)* 1

Of length at least 3 that ends with 00.

RE: $(0+1)^{+}00$

8. ***What is undecidable problem? Define Post's Correspondence Problem with an example.*** *[1+4]*

**Answer:** In computability theory and computational complexity theory, an undecidable problem is a decision problem for which it is proved to be impossible to construct an algorithm that always leads to a correct yes-or-no answer. The halting problem is an example: it can be proven that there is no algorithm that correctly determines whether arbitrary programs eventually halt when run.

**The Post Correspondence Problem (PCP)**

The Post Correspondence Problem (PCP), introduced by Emil Post in 1946, is an undecidable decision problem. The PCP problem over an alphabet ∑ is stated as follows Given the following two lists, M and N of non-empty strings over ∑ :

- $M = (x_1, x_2, x_3, \ldots\ldots\ldots, xn)$
- $N = (y_1, y_2, y_3, \ldots\ldots\ldots, yn)$

Scanned by TapScanner

We can say that there is a Post Correspondence Solution, if for some $i_1, i_2 \ldots ik$, where $1 \le ij \le n$, the condition $xi1 \ldots\ldots xik = yi_1 \ldots\ldots yik$ satisfies.

**Example 1:** Find whether the lists M = (abb, aa, aaa) and N = (bba, aaa, aa) have a Post Correspondence Solution?

**Solution**

| | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| M | abb | aa | aaa |
| N | bba | aaa | aa |

Here, $x_2x_1x_3$ = 'aaabbaaa' and $y_2y_1y_3$ = 'aaabbaaa'

We can see that $x_2x_1x_3 = y_2y_1y_3$

Hence, the solution is i = 2, j = 1, and k = 3.

**Example 2:** Find whether the lists M = (ab, bab, bbaaa) and N = (a, ba, bab) have a Post Correspondence Solution?

**Solution**

| | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| M | ab | bab | bbaaa |
| N | a | ba | bab |

In this case, there is no solution because $|x_2x_1x_3| \neq |y_2y_1y_3|$ (Lengths are not same)

Hence, it can be said that this Post Correspondence Problem is undecidable.

9. ***How pumping lemma can be used to prove that any language is not a regular language? Show that language, L = {$0^r$ $1^r$\|n ≥0} is not a regular language.*** *[4+1]*

**Answer: Method to prove that a language L is not regular**

- At first, we have to assume that L is regular.
- So, the pumping lemma should hold for L.
- Use the pumping lemma to obtain a contradiction:
  - ✓ Select w such that $|w| \ge c$
  - ✓ Select y such that $|y| \ge 1$
  - ✓ Select x such that $|xy| \le c$
  - ✓ Assign the remaining string to z.
  - ✓ Select k such that the resulting string is not in L.

Hence L is not regular.

Pumping Lemma is to be applied to show that certain languages are not regular. It should never be used to show a language is regular.

- If L is regular, it satisfies Pumping Lemma.
- If L does not satisfy Pumping Lemma, it is non-regular.



Let L be a regular language. Then there exists a constant 'c' such that for every string w in L:

$|w| \ge c$

We can break w into three strings, w = xyz, such that:

- $|y| > 0$
- $|xy| \le c$
- For all $k \ge 0$, the string $xy^kz$ is also in L.

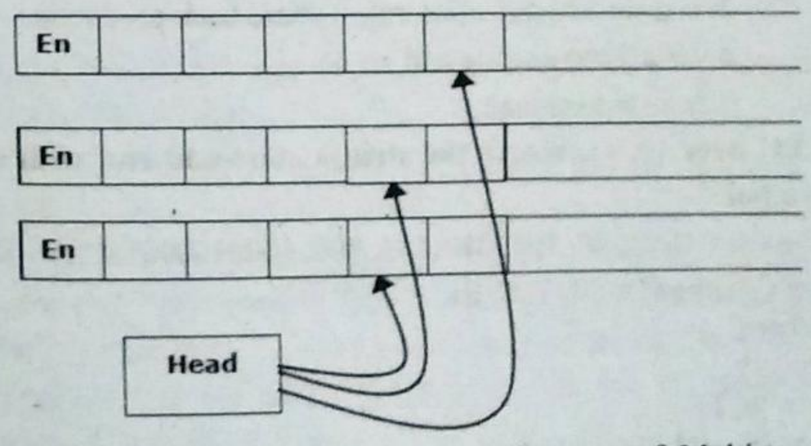
**Show that language, L = {$0^n$ $1^n$\|n ≥0} is not a regular language**

**Answer:**

- At first, we assume that L is regular and n is the number of states.
- Let $w = 0^n1^n$. Thus $|w| = 2n \ge n$.
- By pumping lemma, let w = xyz, where $|xy| \le n$.
- Let $x = 0^p$, $y = 0^q$, and $z = 0^r1^n$, where $p + q + r = n$, $p \neq 0$, $q \neq 0$, $r \neq 0$. Thus $|y| \neq 0$.
- Let k = 2. Then $xy^2z = 0^p0^{2q}0^r1^n$.
- Number of as = $(p + 2q + r) = (p + q + r) + q = n + q$
- Hence, $xy^2z = 0^{n+q}\,1^n$. Since $q \neq 0$, $xy^2z$ is not of the form $0^n1^n$.
- Thus, $xy^2z$ is not in L. Hence L is not regular.

10. ***Discuss how Turing Machine with multiple tracks differs from a Turing Machine with multiple tapes.***

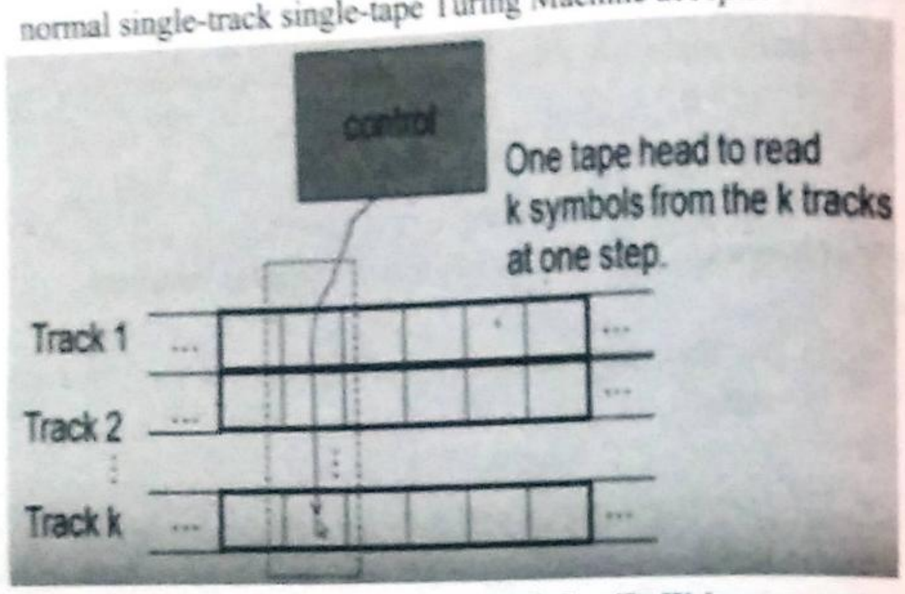
**Answer:** Multi-tape Turing Machines have multiple tapes where each tape is accessed with a separate head. Each head can move independently of the other heads. Initially the input is on tape 1 and others are blank. At first, the first tape is occupied by the input and the other tapes are kept blank. Next, the machine reads consecutive symbols under its heads and the TM prints a symbol on each tape and moves its heads.




Multi-track Turing machines, a specific type of Multi-tape Turing machine, contain multiple tracks but just one tape head reads and

Scanned by TapScanner

writes on all tracks. Here, a single tape head reads n symbols from n tracks at one step. It accepts recursively enumerable languages like a normal single-track single-tape Turing Machine accepts.




11. ***How context free grammars are defined? Write a context free grammar over {0, 1}, where the strings start and end with the same symbol.*** *[2+3]*

**Answer:**

Context-free grammars (CFGs) are used to describe context-free languages. A context-free grammar is a set of recursive rules used to generate patterns of strings. A context-free grammar can describe all regular languages and more, but they cannot describe all possible languages.

A context-free grammar can be described by a four-element tuple (V, T, P, S) where

- V is a finite set of variables (which are non-terminal);
- T is a finite set of terminal symbols;
- P is a set of production rules where each production rule maps a variable to a string and
- S is a start symbol.

**CFG over {0, 1}, where the strings start and end with the same symbol**

L = {ε, 1, 0, 11, 00, 101, 010, 111, 000, 10001, 00010,...}

Let CFG be G = (V, T, P, S)

Where,

V={S}

T = {0, 1}

P = {S→0|1|0S0|1S1|ε}

And S=S



12. ***What is halting problem? How can you argue that halting problem is undecidable?*** *[1+4]*

**Answer:** In computability theory, the halting problem is the problem of determining, from a description of an arbitrary computer program and an input, whether the program will finish running (i.e., halt) or continue to run forever. Alan Turing proved in 1936 that a general algorithm to solve the halting problem for all possible program-input pairs cannot exist. A key part of the proof was a mathematical definition of a computer and program, which became known as a Turing machine; the halting problem is undecidable over Turing machines. Turing's proof is one of the first cases of decision problems to be concluded. The theoretical conclusion of not solvable is significant to practical computing efforts, defining a class of applications which no programming invention can possibly perform perfectly.

The halting problem is a decision problem about properties of computer programs on a fixed Turing-complete model of computation, i.e., all programs that can be written in some given programming language that is general enough to be equivalent to a Turing machine. The problem is to determine, given a program and an input to the program, whether the program will eventually halt when run with that input. In this abstract framework, there are no resource limitations on the amount of memory or time required for the program's execution; it can take arbitrarily long and use an arbitrary amount of storage space before halting. The question is simply whether the given program will ever halt on a particular input.

For example, in pseudocode, the program

while (true) continue

Does not halt; rather, it goes on forever in an infinite loop. On the other hand, the program

print "Hello, world!"

Does halt.

While deciding whether these programs halt is simple, more complex programs prove problematic. One approach to the problem might be to run the program for some number of steps and check if it halts. But if the program does not halt, it is unknown whether the program will eventually halt or run forever.

Scanned by TapScanner

# MODEL QUESTIONS-ANSWERS SET II

Course Title: Theory of Computation — Full Marks: 60
Course No: CSC257 — Pass Marks: 24
Level: B. Sc CSIT Second Year/ Fourth Semester — Time: 3 Hrs

## Section A

**Long Answer Questions**

**Attempt any two questions.** **[2*10=20]**

1. ***Define DFA. How it is differ from NFA? Draw a DFA over {a, b} that accepts the strings ending with abb. Also draw NFA over {a, b} that accepts strings having aa as substring.*** *[2+4+4]*

**Answer**: Definition of Deterministic finite state automata (DFA)

A deterministic finite automaton (DFA) also known as deterministic finite acceptor (DFA), deterministic finite state machine (DFSM), or deterministic finite state automaton (DFSA)—is a finite-state machine that accepts or rejects strings of symbols and only produces a unique computation (or run) of the automaton for each input string. A deterministic finite automaton is defined by a quintuple (5-tuple) as (Q, Σ, δ, q0, F)

Where,

Q = Finite set of states,

Σ = Finite set of input symbols,

δ = A transition function that maps Q × S → Q

q0 = A start state; q0 ∈ Q

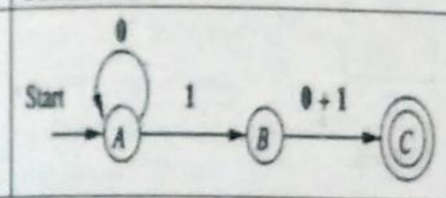
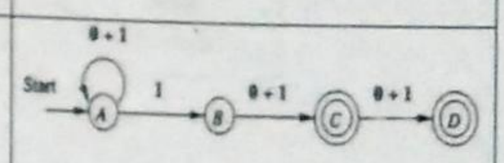
F = Set of final states; F ⊆ Q.

A transaction function δ that takes as arguments a state and an input symbol and returns a state. In our diagram, δ is represented by arcs between states and the labels on the arcs.

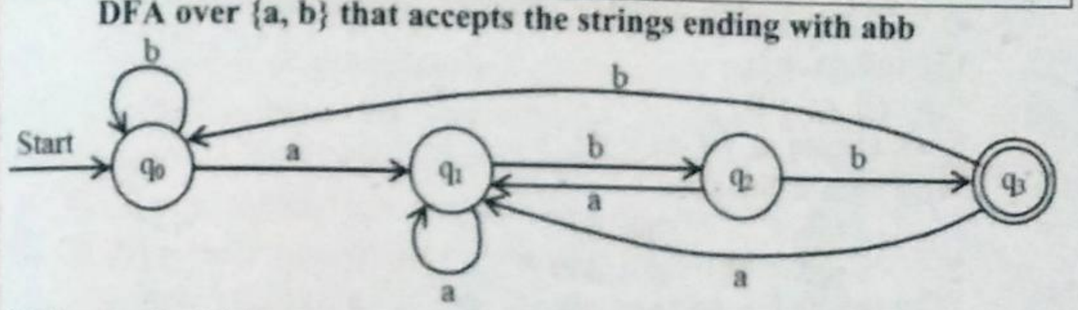
**Difference between Deterministic Finite Automata and the Non deterministic Finite Automata ((DFA Vs NFA):**

| S.N. | DFA | NFA |
|---|---|---|
| 1. | For Every symbol of the alphabet, there is only one state transition in DFA. | We do not need to specify how does the NFA react according to some symbol. |
| 2. | DFA cannot use Empty String transition. | NFA can use Empty String transition. |
| 3. | DFA can be understood as one machine. | NFA can be understood as multiple little machines computing at the same time. |
| 4. | DFA will reject the string if it ends at other than accepting state. | If all of the branches of NFA dies or rejects the string, we can say that NFA reject the string. |

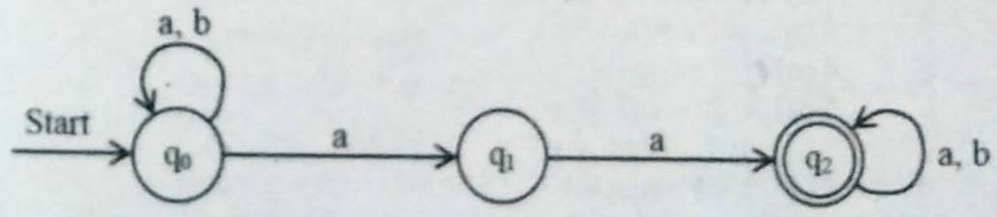


| | | |
|---|---|---|
| 5. | Backtracking is allowed in DFA. | Backtracking is not always allowed in NFA. |
| 6. | DFA can be understood as one machine. | NFA can be understood as multiple little machines computing at the same time. |
| 7. | DFA will reject the string if it end at other than accepting or final state. | If all of the branches of NFA dies or rejects the string, we can say that NFA reject the string. |
| 8. | DFA is more difficult to construct. | NFA is easier to construct. |
| 9. |  |  |

**DFA over {a, b} that accepts the strings ending with abb**




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




2. ***Define pushdown automaton? Construct a PDA for accepting strings of balanced ordered parenthesis. Show instantaneous description of trace for the string [{ }]*** *[2+4+4]*

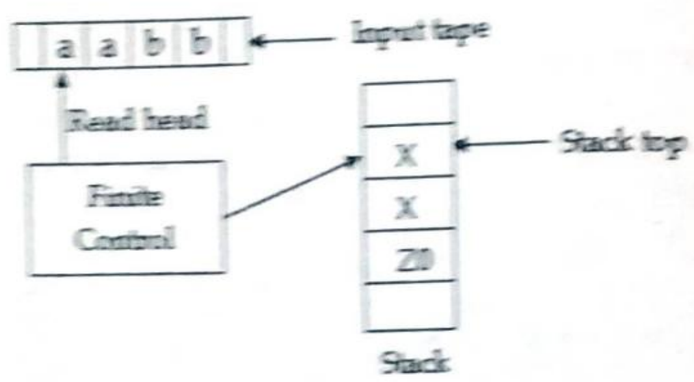
**Answer:** Pushdown Automata is a finite automata with extra memory called stack which helps Pushdown automata to recognize Context Free Languages. A Pushdown Automata (PDA) can be formally defined as follows:

$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ where,

- Q is the set of states
- Σ is the set of input symbols
- Γ is the set of pushdown symbols (which can be pushed and popped from stack)
- $q_0$ is the initial state
- $Z_0$ is the initial pushdown symbol (which is initially present in stack)
- F is the set of final states

Scanned by TapScanner

- $\delta$ is a transition function which maps $Q \times \{\Sigma \cup \epsilon\} \times \Gamma$ into $Q \times \Gamma^*$. In a given state, PDA will read input symbol and stack symbol (top of the stack) and move to a new state and change the symbol of stack.




A DPDA accepting strings of balanced ordered parenthesis P;

Q = {q0, q1, q2},

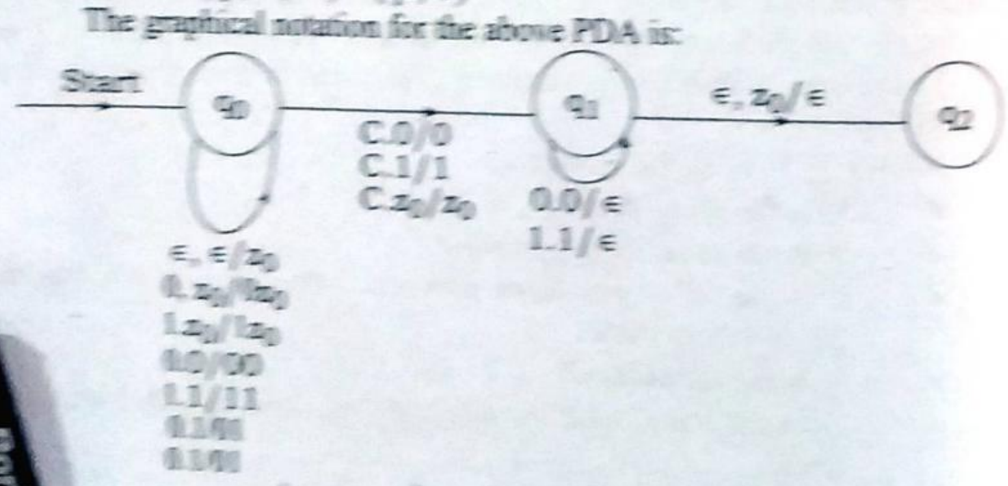
S = {(,),[,],{,}},

Γ = S U {z0}, d, q0, z0, {q2}

Where δ is defined as:

1. δ (q0, ∈, ∈) = (q0, z0)
2. δ (q0, (, z0) = (q1, (z0)
3. δ (q0, [, z0) = (q1, [z0)
4. δ (q0, {, z0) = (q1, {z0)
5. δ (q1, (, ( ) = (q1, (( )
6. δ (q1, [, [ ) = (q1, [[ )
7. δ (q1, {, { ) = (q1, {{ )
8. δ (q1, ), ( ) = (q1, ∈)
9. δ (q1, ], [ ) = (q1, ∈)
10. δ (q1, }, { ) = (q1, ∈)
11. δ (q1, ), { ) = (q1, ({ )
12. δ (q1, (, [ ) = (q1, ([ )
13. δ (q1, ∈, z0) = (q2, ∈)

The graphical notation for the above PDA is:






Evaluate or trace for the string [{ }]

(q0, [{ }], z0) = (q0, { }], [z0)

= (q0, }], {[z0)

= (q1, ], [z0)

= (q2, ∈, z0)

= (q2, ∈, ∈) Accepted

3. ***Difference between TM and Other Automata (FSA and PDA). Construct a Turing Machine accepting the language, L = {0ⁿ1ⁿ | n >= 1}. Also show the transition diagram of the machine. Illustrate whether a string 0011 is accepted by the Turing Machine or not.*** *[2+6+2]*

**Answer:** The most significant difference between the TM and the simpler machine (FSA or PDA) is that; in a Turing Machine, processing a string is no longer restricted to a single left to right pass through input. The tape head can move in both directions and erase or modify any symbol it encounters. The machine can examine part of the input, modify it, take time execute some computation in a different area of the tape, return to re-examine the input, repeat any of these actions and perhaps stop the processing before it has locked at all input.

**Turing Machine accepting the language, L= {0ⁿ1ⁿ | n >= 1}**

- Given finite sequence of 0's and 1's on its tape preceded and followed by blanks.
- The TM will change 0 to an X and then a 1 to Y until all 0's and 1's are matches.
- Starting at left end of the input, it repeatedly changes a 0 to an X and moves to the right over whatever 0's and Y's it sees until comes to a 1.
- It changes 1 to a Y, and moves left, over Y's and 0's until it finds X. At that point, it looks for a 0 immediate to the right. If finds one 0 then changes it to X and repeats the process changing a matching 1 and Y.

Now, TM will have;

M = ({q0, q1, q2, q3, q4}, {0, 1}, {0, 1, X, Y, B}, δ, q0, B, {q4})

The transition rule for the move of M is described by following transition table:

| | 0 | 1 | X | Y | B |
|---|---|---|---|---|---|
| q0 | (q1, X, R) | | | (q3, Y, R) | |
| q1 | (q1, 0, R) | (q2, Y, L) | | (q1, Y, R) | |
| q2 | (q2, 0, L) | | (q0, X, R) | (q2, Y, L) | |
| q3 | | | | (q3, Y, R) | |
| q4 | | | | | (q4, B, R) |

Scanned by TapScanner

Now, the acceptance of 0011 by the TM, M1 is described by following sequence of moves;

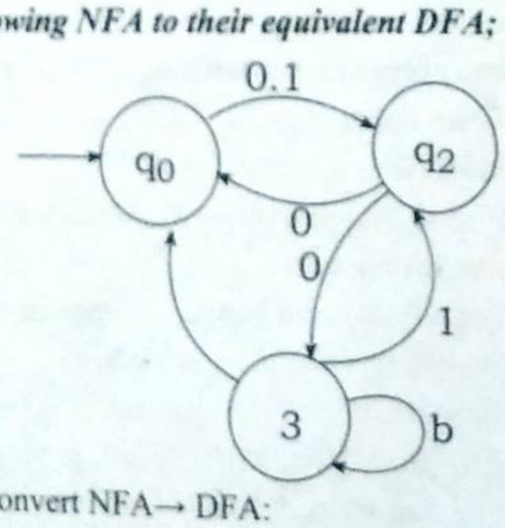
$q_0$ 0011 $\vdash$ X $q_1$011
$\vdash$X0$q_1$11
$\vdash$X$q_2$0Y1
$\vdash$X$q_0$0Y1
$\vdash$XX$q_1$Y1
$\vdash$XXY$q_1$1
$\vdash$XX$q_2$YY
$\vdash$X$q_2$XYY
$\vdash$XX$q_0$YY
$\vdash$XXY$q_3$Y
$\vdash$XXYY$q_3$B
$\vdash$XXYYB$q_4$B Halt and accept.

## Section B

**Short Answer Questions**

**Attempt any eight questions.** **[8×5=40]**

4. ***How to convert NFA to DFA? Write down the conversion steps. Also convert following NFA to their equivalent DFA;*** *[1+4]*




**Answer:** Steps to convert NFA→ DFA:

1. Construct the transaction table of given NFA machine.
2. Scan the next states column in the transaction table from initial state to final state.
3. If any of the next state consists more than one state on the single input alphabet. Then merge them and make it new state. Place this new constructed state in DFA transaction table as present state.
4. The next state of this new constructed state on input alphabet will be the summation of each next state which parts in the NFA transaction table.



5. Repeat step 2 to step 4 until all the states in NFA transaction table will be scanned completely.
6. The finial transaction table must have single next state at single input alphabet.

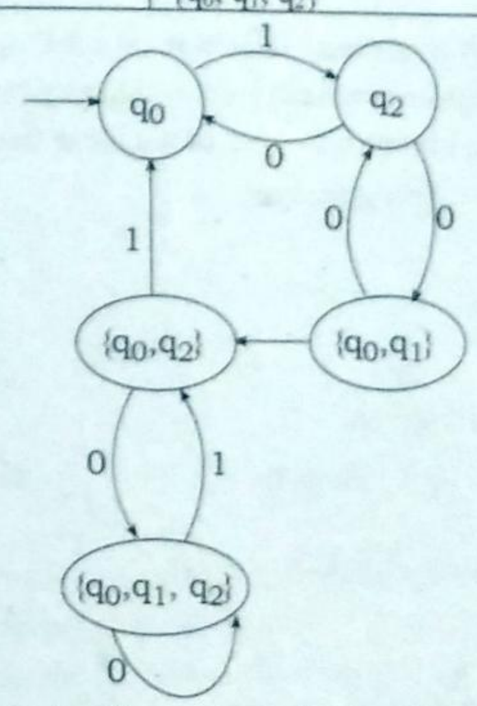
**Construct the equivalent DFA of the NFA given DFA.**

**Step 1:** Transaction Table of NFA from given DFA is;

| Present State | Next State | |
|---|---|---|
| | 0 | 1 |
| →q0 | {q2} | Φ |
| q1 | Φ | {q0, q2} |
| q2* | {q0, q1} | {q0} |

**Step 2:** Transaction Table of DFA:

| Present State | Next State | |
|---|---|---|
| | 0 | 1 |
| →{$q_0$} | {$q_2$} | Φ |
| {$q_2$} | {$q_0$, $q_1$} | {$q_0$} |
| {$q_0$, $q_1$} | {$q_2$} | {$q_0$, $q_2$} |
| {$q_0$, $q_2$} | {$q_0$, $q_1$, $q_2$} | {$q_0$} |
| {$q_0$, $q_1$, $q_2$} | {$q_0$, $q_1$, $q_2$} | {$q_0$} |

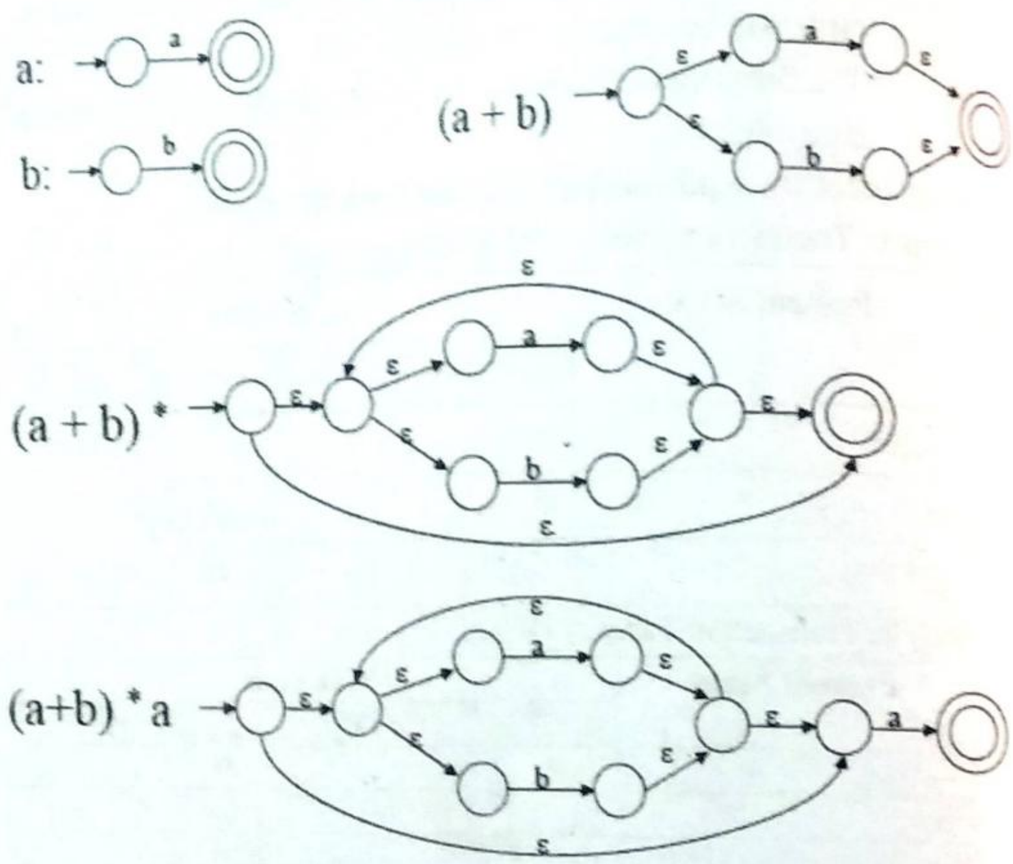



5. ***Define ε-NFA. Configure equivalent epsilon NFA for the regular expression (a + b) * a*** *[1+4]*

**Answer:** In NFA if a transition made without any input symbol is called ε-NFA. Here we need ε-NFA because the regular expressions are easily convertible to ε-NFA.

Rorito Lyk

Scanned by TapScanner




6. ***What is Parse Trees? How Ambiguity of a grammar is detected?*** *[3+2]*

**Answer:** A parse tree is a graphical depiction of a derivation. It is convenient to see how strings are derived from the start symbol. The start symbol of the derivation becomes the root of the parse tree.

**Example:** Let's take a grammar;

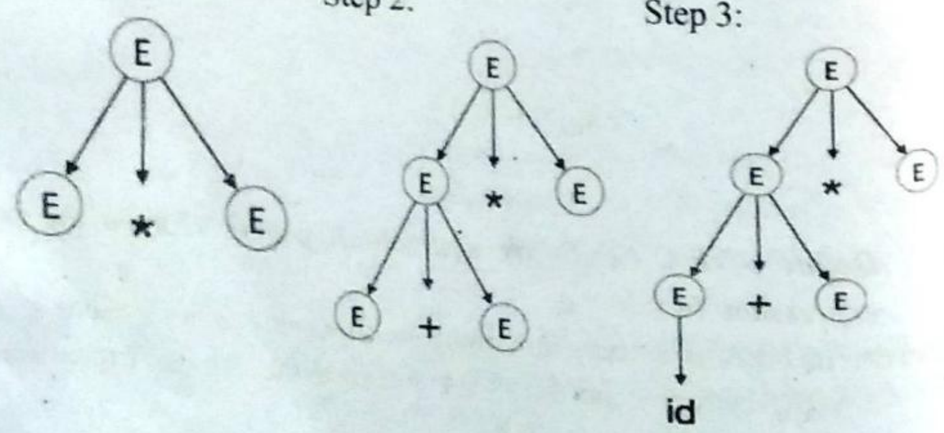
E → E + E

E → E * E

E → id

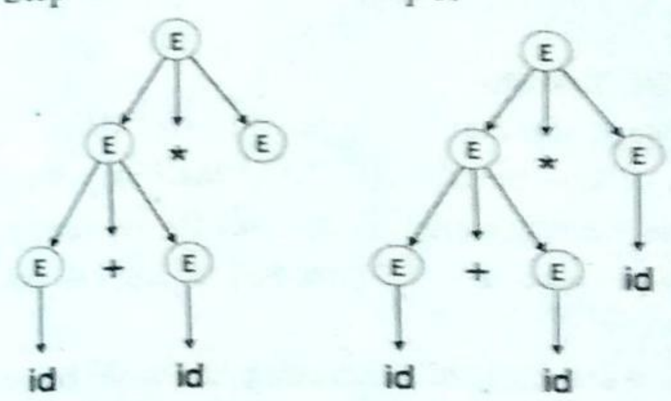
**Input string**: id + id * id

Step 1: Step 2: Step 3:





Step 4: Step 5:



**Ambiguity of a grammar**

A grammar G is said to be ambiguous if it has more than one parse tree (left or right derivation) for at least one string. Also A grammar G is said to be ambiguous if there is a string w ∈ L(G) for which we can construct more than one parse tree rooted at start symbol of the production.

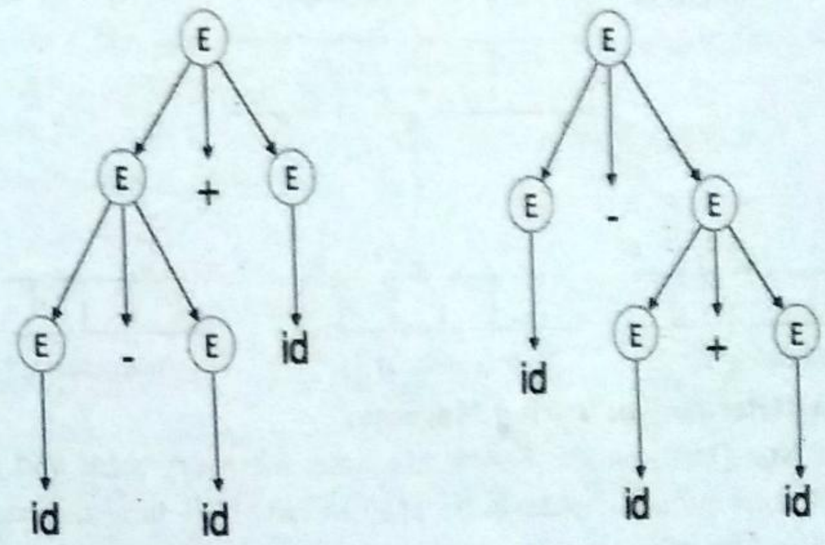
**Example**

E → E + E

E → E – E

E → id

For the string id + id – id, the above grammar generates two parse trees:



7. ***Write the regular expression over {0, 1} for strings*** *[2.5+2.5]*
   - ***all string that begins with a '1' and ends with a '0'***
   - ***string having at most two 0's***

**Answer:** RE for String that begins with a '1' and ends with a '0'.

1(1+0)*0

RE for String that having at most two 0's

1*(0+ ε)1*(0+ ε) 1*

Scanned by TapScanner

8. ***What is Universal Turing Machine? Define Non-Deterministic Turing Machine.*** *[2+3]*

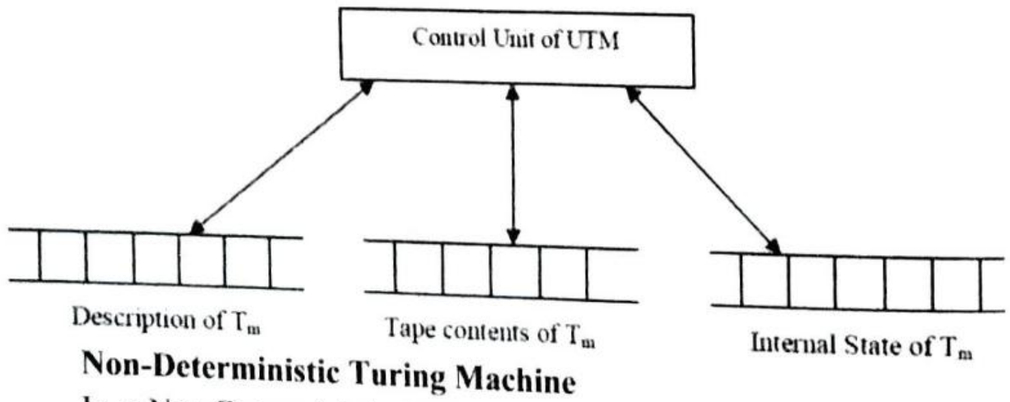
**Answer: Universal Turing Machine**

In computer science, a universal Turing machine (UTM) is a Turing machine that can simulate an arbitrary Turing machine on arbitrary input. The universal machine essentially achieves this by reading both the description of the machine to be simulated as well as the input thereof from its own tape.

If a TM really is a sound model of computation, it should be possible to demonstrate that it can act as a stored program machine, where the program is regarded as an input, rather than hard-wired. We shall construct a Turing Machine Mu, that takes as input a description of a Turing Machine M and an input word x, and simulates the computation of M on input x. A machine such as Mu that can simulate the behavior of an arbitrary TM is called a Universal Turing Machine. Thus, we can describe a Universal Turing Machine Tu as a TM, that on input <M, w>; where M is a TM and w is string of input alphabets, simulates the computation of M on input w.

Specially,

- Tu accepts <M, w> iff M accepts w.
- Tu rejects <M, w> iff M rejects w.




**Non-Deterministic Turing Machine**

In a Non-Deterministic Turing Machine, for every state and symbol, there are a group of actions the TM can have. So, here the transitions are not deterministic. The computation of a non-deterministic Turing Machine is a tree of configurations that can be reached from the start configuration.

An input is accepted if there is at least one node of the tree which is an accept configuration, otherwise it is not accepted. If all branches of the computational tree halt on all inputs, the non-deterministic Turing Machine is called a Decider and if for some input, all branches are rejected, the input is also rejected.

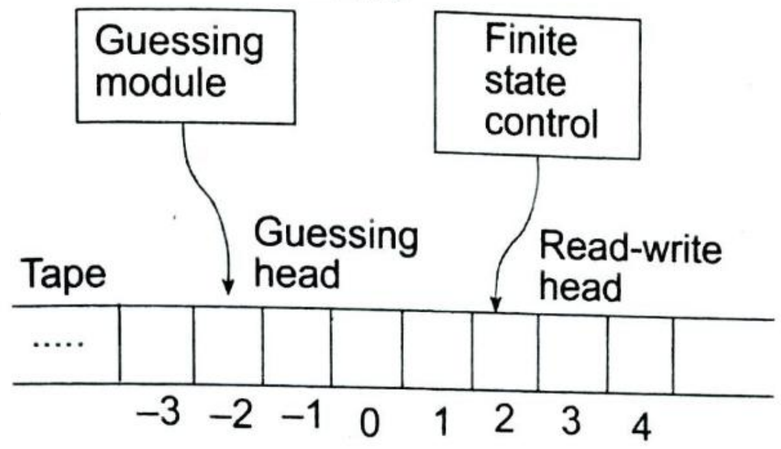


A non-deterministic Turing machine can be formally defined as a 6-tuple (Q, X, $\sum$, $\delta$, $q_0$, B, F) where,

- **Q** is a finite set of states
- **X** is the tape alphabet
- $\sum$ is the input alphabet
- **$\delta$** is a transition function;

$\delta$: Q × X → P(Q × X × {Left_shift, Right_shift}).

- **$q_0$** is the initial state
- **B** is the blank symbol
- **F** is the set of final states




9. ***What is Chomsky normal form? Write down algorithm to convert CFG to CNF. Also*** **convert the following CFG into CNF.** *[1+2+2=5]*
**S → ASA | aB, A → B | S, B → b | ε**

**Answer:** A CFG is in Chomsky Normal Form if the Productions are in the following forms:

- A → a
- A → BC
- S → ε

Where A, B, and C are non-terminals and **'a'** is terminal.

Algorithm to Convert into Chomsky Normal Form:

- Step 1 – If the start symbol S occurs on some right side, create a new start symbol S' and a new production S'→ S.
- Step 2 – Remove Null productions. (Using the Null production removal algorithm discussed earlier)
- Step 3 – Remove unit productions. (Using the Unit production removal algorithm discussed earlier)
- Step 4 – Replace each production A → B1...Bn where n > 2 with A → B1Cwhere C → B2 ...Bn. Repeat this step for all productions having two or more symbols in the right side.

Scanned by TapScanner

- Step 5 – If the right side of any production is in the form A → aB where a is a terminal and A, B are non-terminal, then the production is replaced by A → XB and X → a. Repeat this step for every production which is in the form A → aB.

**Problem:** Convert the following CFG into CNF

S → ASA | aB, A → B | S, B → b | ε

**Solution:**

(1) Since S appears in R.H.S, we add a new state S0 and S0→S is added to the production set and it becomes:

S0→S, S→ ASA | aB, A → B | S, B → b | ∈

(2) Now we will remove the null productions

B → ∈ and A → ∈

After removing B → ε, the production set becomes –

S0→S, S→ ASA | aB | a, A → B | S | ∈, B → b

After removing A → ∈, the production set becomes –

$S_0$→S, S→ ASA | aB | a | AS | SA | S, A → B | S, B → b

(3) Now we will remove the unit productions.

After removing S → S, the production set becomes –

$S_0$→S, S→ ASA | aB | a | AS | SA, A → B | S, B → b

After removing $S_0$→ S, the production set becomes –

$S_0$→ ASA | aB | a | AS | SA, S→ ASA | aB | a | AS | SA

A → B | S, B → b

After removing A→ B, the production set becomes –

S0 → ASA | aB | a | AS | SA, S→ ASA | aB | a | AS | SA

A → S | b

B → b

After removing A→ S, the production set becomes –

S0 → ASA | aB | a | AS | SA, S→ ASA | aB | a | AS | SA

A → b |ASA | aB | a | AS | SA, B → b

(4) Now we will find out more than two variables in the R.H.S

Here, S0→ ASA, S → ASA, A→ ASA violates two Non-terminals in R.H.S.

Hence we will apply step 4 and [illegible] 5 to get the following final production set which is in CNF:

S0→ AX | aB | a | AS | SA

S→ AX | aB | a | AS | SA

A → b |AX | aB | a | AS | SA

B → b

X → SA



(5) We have to change the productions S0→ aB, S→ aB, A→ aB

And the final production set becomes:

S0→ AX | YB | a | AS | SA

S→ AX | YB | a | AS | SA

A → b A → b |AX | YB | a | AS | SA

B → b

X → SA

Y → a

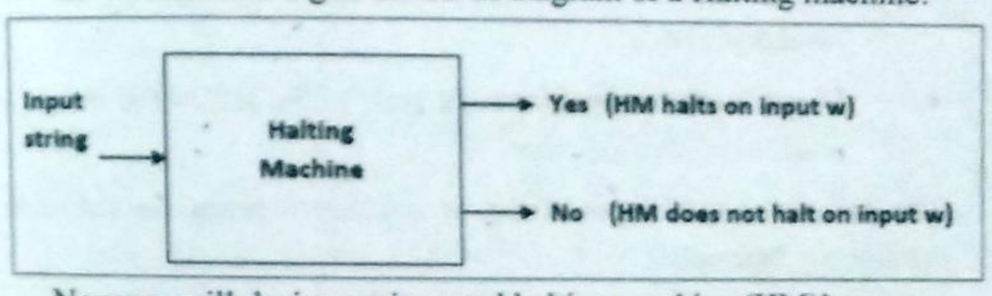
**10. Describe Turing Machine Halting Problem with example. [5]**

**Answer:**

**Input** – A Turing machine and an input string w.

Problem – Does the Turing machine finish computing of the string w in a finite number of steps? The answer must be either yes or no.

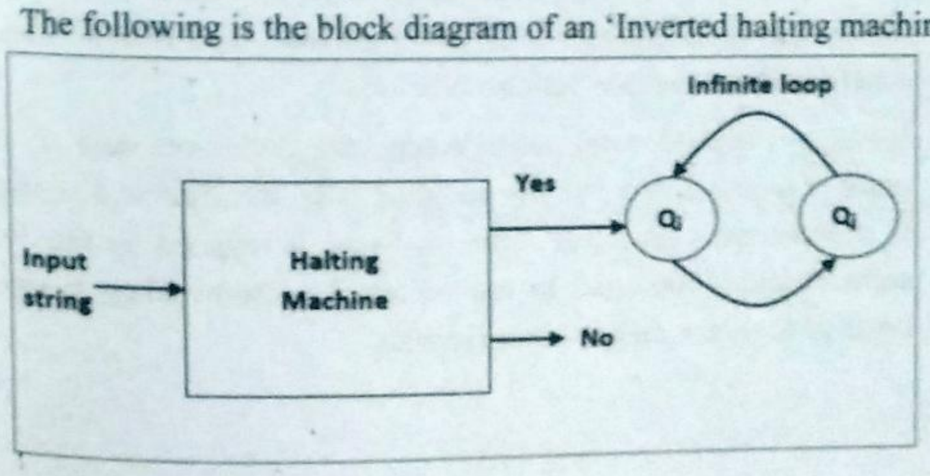
**Proof** – At first, we will assume that such a Turing machine exists to solve this problem and then we will show it is contradicting itself. We will call this Turing machine as a **Halting machine** that produces a 'yes' or 'no' in a finite amount of time. If the halting machine finishes in a finite amount of time, the output comes as 'yes', otherwise as 'no'. The following is the block diagram of a Halting machine:




Now we will design an inverted halting machine (HM)' as –

- If H returns YES, then loop forever.
- If H returns NO, then halt.

The following is the block diagram of an 'Inverted halting machine' –




Scanned by TapScanner

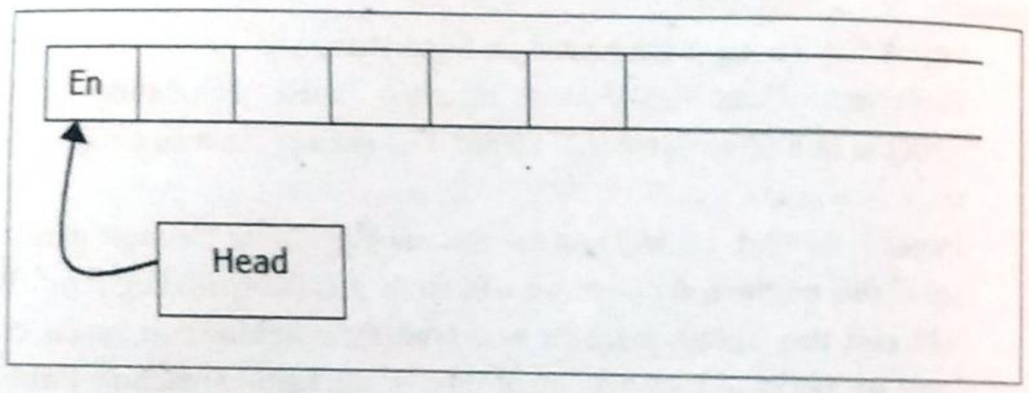
Further, a machine $(HM)_2$ which input itself is constructed as follows:

- If $(HM)_2$ halts on input, loop forever.
- Else, halt.

Here, we have got a contradiction. Hence, the halting problem is undecidable.

*11.* ***efine Semi-Infinite Tape Turing Machine.*** *[5]*

**Answer:** A Turing Machine with a semi-infinite tape has a left end but no right end. The left end is limited with an end marker.




It is a two-track tape:

- Upper track: It represents the cells to the right of the initial head position. .
- Lower track: It represents the cells to the left of the initial head position in reverse order.

The infinite length input string is initially written on the tape in contiguous tape cells.

The machine starts from the initial state q0 and the head scans from the left end marker 'End'. In each step, it reads the symbol on the tape under its head. It writes a new symbol on that tape cell and then it moves the head either into left or right one tape cell. A transition function determines the actions to be taken.

It has two special states called accept state and reject state. If at any point of time it enters into the accepted state, the input is accepted and if it enters into the reject state, the input is rejected by the TM. In some cases, it continues to run infinitely without being accepted or rejected for some certain input symbols.



*12.* ***Write down the languages (Regular Set) for following regular expressions.*** *[5]*

***Regular Expressions***

- ***(0 + 10*)***
- ***(0*10*)***
- ***(0 + ε)(1 + ε)***
- ***(a+b)****
- ***(a+b)*abb***
- ***(11)****
- ***(aa)*(bb)*b***
- ***(aa + ab + ba + bb)****

**Answer:**

| Regular Expressions | Regular Set |
|---|---|
| (0 + 10*) | L = { 0, 1, 10, 100, 1000, 10000, ... } |
| (0*10*) | L = {1, 01, 10, 010, 0010, ...} |
| (0 + ε)(1 + ε) | L = {ε, 0, 1, 01} |
| (a+b)* | Set of strings of a's and b's of any length including the null string. So L = { ε, a, b, aa , ab , bb , ba, aaa.....} |
| (a+b)*abb | Set of strings of a's and b's ending with the string abb. So L = {abb, aabb, babb, aaabb, ababb...} |
| (11)* | Set consisting of even number of 1's including empty string, So L= {ε, 11, 1111, 111111, .....} |
| (aa)*(bb)*b | Set of strings consisting of even number of a's followed by odd number of b's , so L = {b, aab, aabbb, aabbbbb, aaaab, aaaabbb, .....} |
| (aa + ab + ba + bb)* | String of a's and b's of even length can be obtained by concatenating any combination of the strings aa, ab, ba and bb including null, so L = {aa, ab, ba, bb, aaab, aaba, .....} |

Scanned by TapScanner

# Model Questions Sets For Practice

## MODEL SET 1

Course Title: Theory of Computation — Full Marks: 60
Course No: CSC257 — Pass Marks: 24
Level: B. Sc CSIT Second Year/ Fourth Semester — Time: 3 Hrs

### Section A

**Long Answer Questions**

**Attempt any two questions.** **[2*10=20]**

1. Define the extended transition function of DFA. Construct a DFA equivalent to the NFA. M= ({p, q, r},{0, 1}, δ, p,{q, s}) Where δ is defined in the [2+8=10]

| | 0 | 1 |
|---|---|---|
| P | {q, s} | {q} |
| Q | {r} | {q, r} |
| R | {s} | {p} |
| S | - | {p} |

2. Explain the construction of NFA with ε-transition from any given regular expression.
   Let A = (Q, ∑, δ, $q_0$, qf ) be a DFA and suppose that for all a in ∑we have δ($q_0$, a)= δ(qf ,a). Show that if x is a non empty string in L(A), then for all k > 0,xk is also in L(A).
3. **Explain in detail notes on Turing Machine with example.** Construct a Turing Machine accepting Palindrome. Also show the transition diagram of the machine. Illustrate whether a string bbbaabbb is accepted by the Turing Machine or not. [2+6+2]

### Section B

**Short Answer Questions**

**Attempt any eight questions.** **[8*5=40]**

4. Simplify the following grammar
   S→aAa | bBb | BB
   A→C
   B→S | A
   C →S | ϵ
5. What are the closure properties of CFL? State the proof for any two properties. [1+4]



6. What is additional feature PDA has when compared with NFA? Is PDA superior over NFA in the sense of language acceptance? Justify your answer.
7. Define instantaneous description of a PDA.
8. What is un-decidable problem? Define Post's Correspondence Problem with an example. [1+4]
9. How pumping lemma can be used to prove that any language is not a regular language? Show that language, L = {$0^n$ $1^n$|n ≥0} is not a regular language. [4+1]
10. Discuss how Turing Machine with multiple tracks differs from a Turing Machine with multiple tapes. [5]
11. How context free grammars are defined? Write a context free grammar over {0,1}, where the strings start and end with the same symbol. [2+3]
12. What is halting problem? How can you argue that halting problem is un-decidable? [1+4]

## MODEL SET 2

### Section A

**Long Answer Questions**

**Attempt any two questions.** **[2*10=20]**

1. Is it true that the language accepted by any NFA is different from the regular language? Justify your Answer. Consider the following ε–NFA: Compute the ε–closure of each state and find it's equivalent DFA. (2+1+7)

| | E | A | b | C |
|---|---|---|---|---|
| P | {q} | {p} | Φ | Φ |
| Q | {r} | Φ | {q} | Φ |
| R | Φ | Φ | Φ | {r} |

2. Describe the applications of Pumping lemma. Prove that L= {$a^n$ $b^n$ for n=0, 1, 2, 3……} is not regular.
3. Define push down automata. Also describe the applications of push down automata. Design PDA for the grammar G= (V, T, P, S) where V={S}, T={a, b, c} and P is defined as,
   S→a S a
   S→ b S b
   S→ c

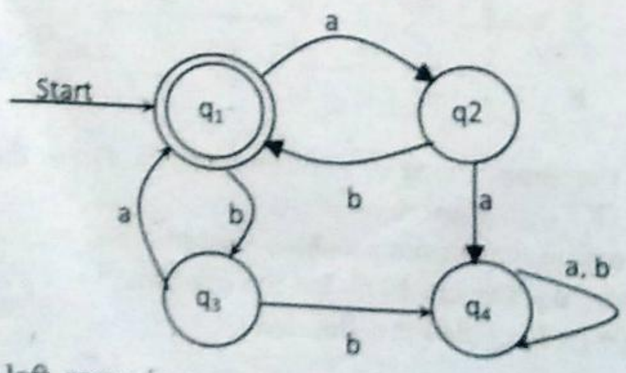
Scanned by TapScanner

## Section B

**Short Answer Questions**

**Attempt any eight questions.** **[8*5=40]**

4. Consider the following grammar
   S→AB
   A→a
   B→C/b
   C→D
   D→E
   E→a
5. Define DFA? How it is differ from NFA? Design a finite automata which accepts the language L={w/w has both an even number of 0's and an even number of 1's over alphabet ∑={0, 1}.
6. Describe universal Turing machine with suitable example.
7. Design a Turing machine that accepts the language of all strings which contain aba as a substring.
8. Define class P, NP and NP complete problems with suitable practical examples.
9. Define Chomsky normal form with suitable example. How to convert CFG to CNF? Change the following grammar into CNF
   S→1A/0B
   A→1AA/0S/0
   B→0BB/1
10. Let G be CFG
    S→bB/aA
    A→b/bS/aAA
    B→a/aS/bBB
    For the string bbaababa find
    - Left most derivation
    - Right most derivation and
    - Parse tree
11. Define Ardens's theorem. Use Ardens's theorem find the regular expression of following transition diagram,




What is left recursion grammar? How left recursion eliminate from given grammar? Explain with suitable example.



# MODEL SET 3

## Section A

**Long Answer Questions**

**Attempt any two questions.** **[2*10=20]**

1. Let input alphabet be ∑ = {a, b, c} and L be the language of all words in which all the a's come before the b's and there are same number of a's as b's and arbitrarily many c's that can be in front, behind or among the a's and b's. Some words in L are abc, caabcb, ccacaabcccbccbc.
   - Show the language is not regular
   - Design a PDA for above given language.
2. Define formal definition of Turing machine. How Turing machine differ from Push down automata? Design a Turing machine that recognizes the language of all strings of even length over alphabet ∑ = {a, b}.
3. Define CFG. Design CFG for ∑ = {a, b} that generates the set of
   a. All strings with exactly one a.
   b. All strings with at least one a
   c. All strings with at least 3 a's

## Section B

**Short Answer Questions**

**Attempt any eight questions.** **[8*5=40]**

4. What is regular expression? How it is differ from CFG? Write the regular expression over alphabet ∑ = {a, b, c} containing at least one a and at least one b.
5. Define ε-NFA. How it is differ from NFA? Construct the ε-NFA for the regular expression (0 + 1)* 1 (0 + 1).
6. Define Bacos Naur Form (BNF). Write CFG for the regular expression,
   r = 0* 1 (0 + 1)*
7. What is the use of PDA? Construct a push down automata to accept the following language,
   $L = \{w\, w^R : w \in \{a, b\}^*\}$
8. Describe properties of Turing machine. Design a Turing machine that accepts the language of all strings which contain aba as a substring.
9. What is the advantage of using multi tapes Turing machine over simple Turing machine? Describe universal Turing machine with suitable example.
10. What is context sensitive grammar? How it is differ from context free grammar? Explain their differences with suitable example.
11. Define class P problem with suitable example. How class P problems differ from class NP problems? Explain.
12. Construct the transaction system for the following regular expressions;
    a. r = 1 + 00 + 010*1
    b. r = (ab + c*)*b

Scanned by TapScanner

## MODEL SET 4

### Section A

**Long Answer Questions**

**Attempt any two questions.** **[2*10=20]**

1. Define Pumping lemma for CFG. Prove that language $L = \{ww/w \in \{a, b\}^*\}$ is not context free.
2. Define formal definition of Turing machine. How Turing machine differ from Finite automata? Design a Turing machine that recognizes the language of all strings of odd length over alphabet $\sum = \{a, b\}$.
3. Define CFG. How it is differ from regular expression? Design regular expression for $\sum = \{a, b\}$ that generates the set of
   a. All strings with exactly one a.
   b. All strings with at least one a
   c. All strings with at least 3 a's

### Section B

**Short Answer Questions**

**Attempt any eight questions.** **[8*5=40]**

4. Define Kleen closure operation. How it is differ from positive closure? Find Kleen closure of following expressions over $\sum=\{a, b\}$
   a. $a^*$
   b. $(ab)^*$
   c. $(a + b)^*$
   d. $ab + a^*$
   e. $(ba + ab^*)^*$
5. Define NFA. How it is differ from ε-NFA? Construct NFA over {a, b} that accepts either empty strings or ab, or aab, or aba. Use extended transaction function to determine whether the NFA accepts bbbabb and aab.
6. Construct a Turing machine accepting the language, $L = \{a^n b^{2n} \mid n>=1\}$. Also draw the equivalent transaction diagram.
7. Define CFG. Write CFG for the regular expression,
   $r = (0+1).(0 + 1)^*$
8. What are the applications of PDA? Construct a push down automata for balanced parentheses, i.e. $\sum = \{, [, (, ), ], \}$.
9. What is left recursive grammar? How removal of left recursion from grammar is occurring? Remove left recursion from following grammar;
   $S \rightarrow Sa \mid SBb \mid SAab \mid a \mid b \mid c$
   $A \rightarrow Aab \mid Ac \mid a \mid bb \mid c$
   $B \rightarrow abc \mid a \mid b \mid \varepsilon$



10. Define regular expression. Define regular expression for following statements:
    a. An IPV4 Address
    b. A MAC Address
    c. Valid identifier defined in programming like C
11. What does NP completeness means? Show that SAT is NP-complete.
12. Simplify following grammar and convert to the CNF
    $S \rightarrow abSb \mid aAb \mid a$
    $A \rightarrow bS \mid aAAb$

## MODEL SET 5

### Section A

**Long Answer Questions**

**Attempt any two questions.** **[2*10=20]**

1. Is context free grammar exploring the meaning of the term "Context free" with a suitable example? Write the CFG for the language $L = \{x\, 0^n\, y\, 1^n\, z \mid n>=0\}$, and give the parse tree for the sting x000y11z.
2. Define Turing machine. Design a Turing machine for a language $L= \{a, b, c\}|n > = 0\}$. Show instantaneous description for aabbcc and abcc.
3. Define transaction diagram and transaction table of DFA. Design a FA over alphabet $\sum = \{0, 1\}$, which accepts the set of strings either start with 01 or end with 01.

### Section B

**Short Answer Questions**

**Attempt any eight questions.** **[8*5=40]**

4. Define 'terms', 'language' and 'alphabet'. Find all pairs of set of A and B for which A.B = { ab, aaa, abab, abbb, abaaa, ababbb}.
5. Let us define the alphabet set as $\sum = \{0, 1\}$, then
   a. Build a DFA that accepts all the strings ending in 00.
   b. Build a DFA that accepts the strings that do not have 1110 as their prefix.
6. Show that if $M = (Q, \sum, \delta, q_0, F)$ and $N' = (Q, \sum, \delta, q_0, Q\text{-}F)$ are two DFA's, then $(L(M))'=L(M')$
7. How to convert given regular expression to equivalent ε-NFA? Convert following regular expression to their equivalent ε-NFA,
   a. $(a + b)^*$
   b. b

Scanned by TapScanner

8. Write down the applications of pumping lemma. Prove that $L = \{a^n b a^n \text{ for } n = 0, 1, 2, 3\ldots\}$ is not regular.
9. What is the use of left most derivation and right most derivation? Let G be CFG with production rules,
   $S \rightarrow bB \mid aA$
   $A \rightarrow b \mid bS \mid aAA$
   $B \rightarrow a \mid aS \mid bBB$
   For the string bbaababa find
   a. Left most derivation
   b. Right most derivation and
   c. Parse tree
10. Define notations used in Turing machine. Design a Turing machine that recognizes the set of all strings of 0's and 1's containing at least one 1.
11. Configure a push down automata accepting the language, $L = \{w\,C\,w \mid w \in (a + b)^*\}$. Show instantaneous description of string abbCbba and baCba.
12. What do you mean by decision problem? Give Turing's proof of Halting problem as undecided problem.

## MODEL SET 6

### Section A

**Long Answer Questions**

**Attempt any two questions.** **[2*10=20]**

1. Prove that there exists a DFA for every ε-NFA. Prove that the complement of a regular language is also regular.
2. What are the required fields of an instantaneous description of a Turing machine? Design a Turing machine with no more than three states that accepts the language a(a+b)*.
3. Convert the following grammar into CNF
   a. $S \rightarrow cBA,\ S \rightarrow A,\ A \rightarrow cB,\ A \rightarrow AbbS,\ B \rightarrow aaa$
   b. $S \rightarrow a \mid AAB,\ A \rightarrow ab \mid aB \mid \varepsilon,\ B \rightarrow aba \mid \varepsilon$
   c. $S \rightarrow aAD,\ A \rightarrow aB \mid bAB,\ B \rightarrow b,\ D \rightarrow d$

### Section B

**Short Answer Questions**

**Attempt any eight questions.** **[8*5=40]**

4. Define ε-closure (q) with an example. Construct a DFA for the language over {0, 1}* such that it contains "000" as a substring.
5. State the relations among regular expression, deterministic finite automata, non deterministic finite automaton and finite automaton with epsilon transition. Draw a Non-deterministic finite automaton to accept strings containing the substring 0101.



6. Let us define the alphabet set as $\Sigma = \{0, 1\}$, then
   a. Build a DFA that accepts all the strings ending in 00 and start with 11.
   b. Build a DFA that accepts the strings that do not have 1110 as their prefix.
   c. Build a NFA that contains at least one 0's and one 1's.
7. What are the applications of Turing Machine? Construct a Turing Machine for language $L = \{ww^r \mid w \in \{0, 1\}\}$
8. Prove that if there exists a PDA that accepts by final state then there exists an equivalent PDA that accepts by Null state.
9. Find the context free languages for the following grammars.
   a. $S \rightarrow aSbS \mid bSaS \mid \varepsilon$
   b. $S \rightarrow aSb \mid ab$
   c. $S \rightarrow aSb \mid aAb,\ A \rightarrow bAa,\ A \rightarrow ba$
10. What is ambiguous grammar? Let G be CFG with production rules,
    $S \rightarrow a \mid \wedge \mid (T)$
    $T \rightarrow T,S \mid S$
    Find left most derivation, rightmost derivation and parse tree for (((a, a),^(a)),a)
11. What are the conventional notations of Push down Automata? Convert the following CFG to a PDA,
    $S \rightarrow aAA,\ A \rightarrow aS \mid bS \mid a$
12. State the two normal forms and give an example. Convert the following grammar G in greibach normal form.
    $S \rightarrow ABb \mid a$
    $A \rightarrow aaA \mid B$
    $B \rightarrow bAb$

## MODEL SET 7

### Section A

**Long Answer Questions**

**Attempt any two questions.** **[2*10=20]**

1. List the primary objectives of Turing Machine. Is the language $L = \{a^n b^n c^n \mid n>=1\}$ is context free? Justify.
2. List the closure properties of Context Free Languages. Let G be the grammar $S \rightarrow aB \mid bA$
   $A \rightarrow a \mid aS \mid bAA$
   $B \rightarrow b \mid bS \mid aBB$.
   For the string aaabbabbba, Find
   a. LMD (Left most derivation)
   b. RMD (Right most derivation)

Scanned by TapScanner

3. Specify the use of context free grammar. Convert the following grammar into an equivalent one with no unit productions and no useless symbols
   S→ABA
   A→aAA|aBC|bB
   B→ A|bB|Cb
   C→CC|Cc

**Short Answer Questions**

**Attempt any eight questions.** **[8*5=40]**

4. When we say a problem is decidable? Give an example of un-decidable problem.
5. Is it true that the language accepted by a non deterministic Turing Machine is different from recursively enumerable language?
6. Prove that for two recursive languages L1 and L2 their union and intersection is recursive.
7. Discuss the various techniques for Turing Machine Construction. Write about Multi tape Turing Machines.
8. State and prove the pumping lemma for CFL. What is its main application? Give two examples.
9. Is NPDA (Nondeterministic PDA) and DPDA (Deterministic PDA) equivalent? Illustrate with an example.
10. Is the following grammar is ambiguous? Justify your answer.
    a. E→ E+E |E*E | id
    b. E→ E+E|E*E|(E)|a
11. Construct the PDA accepting the language
    a. $L=\{(ab)^n \mid n>=1\}$ by empty stack.
    b. $L=\{ww^R \mid w \text{ is in } (a+b)^*\}$
12. What are the applications of CFG? Write down the context free grammar for the language $L = \{ a^n b^n \mid n>=1\}$

## MODEL SET 8

### Section A

**Long Answer Questions**

**Attempt any two questions.** **[2*10=20]**

1. Define parse tree with an example. Construct a CFG over {a,b} generating a language consisting of equal number of a's and b's. Is the grammar below ambiguous S→ SS | (S) | S (S) S | E?
2. What are the different types of language accepted by a PDA and define them? When is Push down automata (PDA) said to be deterministic? Construct PDA for the language $L = \{a^n b^m \mid n>m>=0\}$. Also show that acceptance of string aaaabbb.



3. Prove any two closure properties of regular languages. Construct a minimized DFA from the regular expression
   a. (b/a)*baa
   b. 0*(01)(0/111)*

### Section B

**Short Answer Questions**

**Attempt any eight questions.** **[8*5=40]**

4. Which of the following languages is regular? Justify
   a. $L= \{a^n b^m \mid n, m>=1\}$
   b. $L= \{a^n b^n \mid n>=1\}$
   c. $L= \{0^{n2} \mid n \text{ is an integer}, n>=1\}$
5. What is un-ambiguity? Construct the Context free grammar representing the set of palindromes over (0+1)*
6. What do you mean by null production and unit production? Give an example. Consider the following grammar G with productions
   S→ ABC | BaB
   A→aA|BaC|aaa
   B→bBb|a
   C→CA|AC
   Give a CFG with no useless variables that generates the same language.
7. Prove or disprove that the regular languages are closed under concatenation and complement.
8. Compare NFA and PDA. Is NPDA (Nondeterministic PDA) and DPDA (Deterministic PDA) equivalent? Illustrate with an example.
9. What is chomsky normal form? Convert the following grammar into CNF
   S→aAD
   A→aB|bAB
   B→ b
   D→ d
10. Mention the difference between decidable and un-decidable problems. How to prove that the Post Correspondence problem is Un-decidable?
11. Prove that there exists an NFA with ε-transitions that accepts the regular expression γ.
12. Differentiate regular expression and regular language. Give regular expressions for the following
    a. $L_1$ = set of all strings of 0 and 1 ending in 00 and starting with 1
    b. $L_2$ =set of all strings of 0 and 1 beginning with 0 and ending with 1 and contain even length of 0 and 1.

❑❑❑

Scanned by TapScanner

# Operating System

Course Title: Operating Systems
Course No: CSC259
Nature of the Course: Theory + Lab
Semester: IV

Full Marks: 60+ 20+20
Pass Marks: 24+8+8
Credit Hrs: 3

**Course Objectives**

This course introduces concept of computer networking and discuss the different layers of networking model.

**Course Description**

The main objective of this course is to introduce the understanding of the concept of computer networking with its layers, topologies, protocols & standards, IPv4/IPv6 addressing, Routing and Latest Networking Standards.

**Course Details**

**Unit 1: Operating System Overview** (4 Hrs.)

1.1 Definition, Two views of operating system, Evolution of operating system, Types of OS.

1.2 System Call, Handling System Calls, System Programs, Operating System Structures, The Shell, Open Source Operating Systems

**Unit 2: Process Management** (10 Hrs.)

2.1 Process vs Program, Multiprogramming, Process Model, Process States, Process Control Block.

2.2 Threads, Thread vs Process, User and Kernel Space Threads.

2.3 Inter Process Communication, Race Condition, Critical Section.

2.4 Implementing Mutual Exclusion: Mutual Exclusion with Busy Waiting (Disabling Interrupts, Lock Variables, Strict Alteration, Test and Set Lock), Sleep and Wakeup, Semaphore, Monitors, Message Passing.

2.5 Classical IPC problems: Producer Consumer, Sleeping Barber, Dining Philosopher Problem

2.6 Process Scheduling: Goals, Batch System Scheduling (First-Come First-Served, Shortest Job First, Shortest Remaining Time Next), Interactive System Scheduling (Round-Robin Scheduling, Priority Scheduling, Multiple Queues), Overview of Real Time System Scheduling

**Unit 3: Process Deadlocks** (6 Hrs.)

3.1 Introduction, Deadlock Characterization, Preemptable and Non-preemptable Resources, Resource Allocation Graph, Conditions for Deadlock

3.2 Handling Deadlocks: Ostrich Algorithm, Deadlock prevention, Deadlock Avoidance, Deadlock Detection (For Single and Multiple Resource Instances), Recovery From Deadlock (Through Preemption and Rollback)

**Unit 4: Memory Management** (8 Hrs.)

4.1 Introduction, Monoprogramming vs. Multi-programming, Modelling Multiprogramming, Multiprogramming with fixed and variable partitions, Relocation and Protection.

4.2 Memory management (Bitmaps & Linked-list), Memory Allocation Strategies



4.3 Virtual memory: Paging, Page Table, Page Table Structure, Handling Page Faults, TLB's

4.4 Page Replacement Algorithms: FIFO, Second Chance, LRU, Optimal, LFU, Clock, WS-Clock, Concept of Locality of Reference, Belady's Anomaly

4.5 Segmentation: Need of Segmentation, its Drawbacks, Segmentation with Paging (MULTICS)

**Unit 5: File Management** (6 Hrs.)

5.1 File Overview: File Naming, File Structure, File Types, File Access, File Attributes, File Operations, Single Level, two Level and Hierarchical Directory Systems, File System Layout.

5.2 Implementing Files: Contiguous allocation, Linked List Allocation, Linked List Allocation using Table in Memory, Inodes.

5.3 Directory Operations, Path Names, Directory Implementation, Shared Files

5.4 Free Space Management: Bitmaps, Linked List

**Unit 6: Device Management** (6 Hrs.)

6.1 Classification of IO devices, Controllers, Memory Mapped IO, DMA Operation, Interrupts

6.2 Goals of IO Software, Handling IO(Programmed IO, Interrupt Driven IO, IO using DMA), IO Software Layers (Interrupt Handlers, Device Drivers)

6.3 Disk Structure, Disk Scheduling (FCFS, SSTF, SCAN, CSCAN, LOOK, CLOOK), Disk Formatting (Cylinder Skew, Interleaving, Error handling), RAID

**Unit 7: Linux Case Study** (5 Hrs.)

7.1 History, Kernel Modules, Process Management, Scheduling, Inter-process Communication, Memory Management, File System Management Approaches, Device Management Approaches.

**Laboratory Works:**

The laboratory work includes solving problems in operating system. The lab work should include at least;

- Learn basic Linux Commands
- Create process, threads and implement IPC techniques
- Simulate process Scheduling algorithms and deadlock detection algorithms Simulate page replacement algorithms
- Simulate free space management techniques and disk scheduling algorithms.

**Text Books:**

Modern Operating Systems: Andrew S. Tanenbaum, PHI Publication, Third, edition, 2008.

**Reference Books:**

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, "Operating System Concepts", John Wiley & Sons (ASIA) Pvt. Ltd., Seventh edition, 2005.
2. Harvey M. Deitel, Paul J. Deitel, and David R. Choffnes, "Operating Systems", Prentice Hall, Third Edition, 2003.

Scanned by TapScanner

## TRIBHUVAN UNIVERSITY

Institution of Science and Technology

CSIT Second Year/ Fourth Semester
Operating System [CSC259]

Full Marks: 60
Pass Marks: 24
Time: 3 Hrs

***Candidates are required to give their answers in their own words as far as practicable.***

The figures in the margin indicate full marks.

## T.U. MODEL QUESTIONSN-ANSWERS SET I

**Section 'A'**

**Long Answer Questions**

**Attempt any two questions.** **(2 ×10=20)**

**1. What is sleep and wakeup? Demonstrate problem with suitable code snippet and illustration.**

**Solution:**

Sleep is the system call that causes the caller to block, that is suspended until another process wakes it up. The wakeup call has one parameter, the process to be awakened.

Consider a computer with two processes, H, with high priority and L, with low priority. The scheduling rules are such that H is run whenever it is in ready state. At a certain moment, with L in its critical region, H becomes ready to run (e.g., an I/O operation completes). H now begins busy waiting, but since L is never scheduled while H is running, L never gets the chance to leave its critical region, so H loops forever. This situation is sometimes referred to as the priority inversion problem.

There is a popular example called producer consumer problem which is the most popular problem simulating sleep and wake mechanism.

In producer consumer problem, let us say there are two processes, one process writes something while the other process reads that. The process which is writing something is called producer while the process which is reading is called consumer. The code that simulates the sleep and wake mechanism in terms of providing the solution to producer consumer problem is shown below.

```
#define N 100 //maximum slots in buffer
```



```
#define count=0 //items in the buffer
void producer (void)
{
  int item;
  while(True)
  {
    item = produce_item(); //producer produces an item
    if(count == N) //if the buffer is full then the producer will sleep
      Sleep();
    insert_item (item); //the item is inserted into buffer
    countcount=count+1;
    if(count==1) //The producer will wake up the
    //consumer if there is at least 1 item in the buffer
    wake-up(consumer);
  }
}

void consumer (void)
{
  int item;
  while(True)
  {
    {
      if(count == 0) //The consumer will sleep if the buffer is empty.
      sleep();
      item = remove_item();
      countcount = count - 1;
      if(count == N-1) //if there is at least one slot available in the
buffer
      //then the consumer will wake up producer
      wake-up(producer);
      consume_item(item); //the item is read by consumer.
    }
  }
}
```

Scanned by TapScanner

The producer produces the item and inserts it into the buffer. The value of the global variable count got increased at each insertion. If the buffer is filled completely and no slot is available then the producer will sleep, otherwise it keep inserting. On the consumer's end, the value of count got decreased by 1 at each consumption. If the buffer is empty at any point of time then the consumer will sleep otherwise, it keeps consuming the items and decreasing the value of count by 1.

The consumer will be waked up by the producer if there is at least 1 item available in the buffer which is to be consumed. The producer will be waked up by the consumer if there is at least one slot available in the buffer so that the producer can write that.

Well, the problem arises in the case when the consumer got preempted just before it was about to sleep. Now the consumer is neither sleeping nor consuming. Since the producer is not aware of the fact that consumer is not actually sleeping therefore it keep waking the consumer while the consumer is not responding since it is not sleeping. This leads to the wastage of system calls. When the consumer get scheduled again, it will sleep because it was about to sleep when it was preempted. The producer keep writing in the buffer and it got filled after some time. The producer will also sleep at that time keeping in the mind that the consumer will wake him up when there is a slot available in the buffer. The consumer is also sleeping and not aware with the fact that the producer will wake him up. This is a kind of deadlock where neither producer nor consumer is active and waiting for each other to wake them up. This is a serious problem which needs to be addressed.

2. **When page fault occurs and how it is handled? Demonstrate Second Chance, and LRU page replacement algorithm for memory with three frames and following reference string: 1,3,7,4,5,2,3,6,4,5,7,8, 5,1,4**

**Solution:**

A page fault occurs when a program attempts to access data or code that is in its address space, but is not currently located in the system RAM. So when page fault occurs then following sequence of events happens:



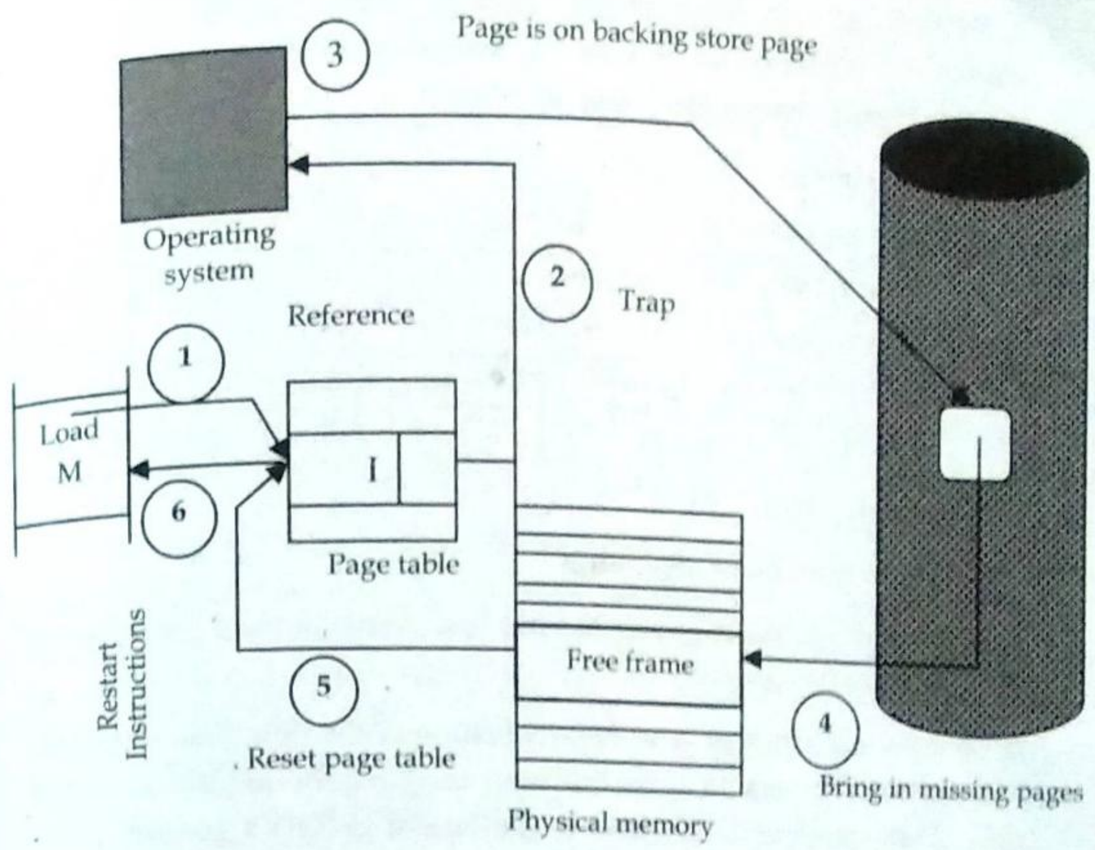



**Figure:** block diagram handling page faults

If a page is needed that was not originally loaded up, then a page fault trap is generated, which must be handled in a series of steps:

- The memory address requested is first checked, to make sure it was a valid memory request.
- If the reference was invalid, the process is terminated. Otherwise, the page must be paged in.
- A free frame is located, possibly from a free-frame list.
- A disk operation is scheduled to bring in the necessary page from disk. (This will usually block the process on an I/O wait, allowing some other process to use the CPU in the meantime)
- When the I/O operation is complete, the process's page table is updated with the new frame number, and the invalid bit is changed to indicate that this is now a valid page reference.
- The instruction that caused the page fault must now be restarted from the beginning, ( as soon as this process gets another turn on the CPU)

**Second Chance Page Replacement Algorithms**

It is the modified form of the FIFO page replacement algorithm. One way to implement is to have a circular queue. If the value is equal to 0, then we proceed to replace the page. But if the reference bit is equal to

Scanned by TapScanner

1, then we give the page second chance. When the page get second chance, its reference bit is cleared. Second-chance page replacement algorithm gives every page a second-chance.

**Reference strings**

| 1 | 3 | 7 | 4 | 5 | 2 | 3 | 6 | 4 | 5 | 7 | 8 | 5 | 1 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1(0) | 1(0) | 1(0) | 4(0) | 4(0) | 4(0) | 3(0) | 3(0) | 3(0) | 5(0) | 3(0) | 3(0) | 5(0) | 5(0) | 5(0) |
| | 3(0) | 3(0) | 3(0) | 5(0) | 5(0) | 5(0) | 6(0) | 6(0) | 6(0) | 7(0) | 7(0) | 7(0) | 1(0) | 1(0) |
| | | 7(0) | 7(0) | 7(0) | 2(0) | 2(0) | 2(0) | 4(0) | 4(0) | 4(0) | 8(0) | 8(0) | 8(0) | 4(0) |
| Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf |

Total page faults = 15

**LRU page replacement algorithm**

In this algorithm, the page that has not been used for the longest period of time has to be replaced.

If we use the recent past as an approximation of the near future, then we will replace the page that has not been used for the longest period of time. This approach is the least-recently-used (LRU) algorithm. LRU replacement associates with each page the time of that page's last use. When a page must be replaced, LRU chooses that page that has not been used for the longest period of time. This strategy is the optimal page replacement algorithm looking backward in time, rather than forward.

**Reference strings**

| 1 | 3 | 7 | 4 | 5 | 2 | 3 | 6 | 4 | 5 | 7 | 8 | 5 | 1 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 4 | 4 | 4 | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 5 | 5 |
| | 3 | 3 | 3 | 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 | 1 | 1 |
| | | 7 | 7 | 7 | 2 | 2 | 2 | 4 | 4 | 4 | 8 | 8 | 8 | 4 |
| Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf | Pf | X | Pf | Pf |

Total page faults = 14

**3. What is I-node? Why it is superior to other file allocation approaches? Consider 20-GB disk with 8-KB block size. How much memory space will be occupied if contiguous, and File allocation table is used for file allocation. Assume that each FAT entry takes 4 byte.**

**Solution:**

In UNIX based operating systems, each file is indexed by an Inode. Inode are the special disk block which is created with the creation of the file system. The number of files or directories in a file system depends on the number of Inodes in the file system. An Inode includes the following information



- Attributes (permissions, time stamp, ownership details, etc) of the file
- A number of direct blocks which contains the pointers to first 12 blocks of the file.
- A single indirect pointer which points to an index block. If the file cannot be indexed entirely by the direct blocks then the single indirect pointer is used.
- A double indirect pointer which points to a disk block that is a collection of the pointers to the disk blocks which are index blocks. Double index pointer is used if the file is too big to be indexed entirely by the direct blocks as well as the single indirect pointer.
- A triple index pointer that points to a disk block that is a collection of pointers. Each of the pointers is separately pointing to a disk block which also contains a collection of pointers which are separately pointing to an index block that contains the pointers to the file blocks.

**Numerical Part,**

Size of disk=20 GB =20 x $2^{20}$ KB = 20971520 KB

Size of block=8 KB

Thus, number of blocks=20971520/8 = 2621440

⇨ Number of Entries in FAT = 2621440

Since size of an entry= 4 byte

⇨ Size of FAT (File Allocation Table)= 2621440 x 4 byte =10485760 byte = 10 MB Ans.

**Short Questions**

**Attempt any eight questions.** **(8 × 5 = 40)**

**4. Define the terms shell and system call? How it is handled? Illustrate with suitable example.**

**Solution: Shell**

The operating system carries out the system calls. Editors, compilers, assemblers, linkers, and command interpreters definitely are not part of the operating system, even though they are important and useful, called the shell. Shell is not part of the operating system, but makes heavy use of many operating system features and thus serves as a good example of how the system calls can be used. It is also the primary interface between a user and the operating system, unless the user is using a graphical user interface. Examples of some shells are *sh, csh, ksh,* and *bash.*

Scanned by TapScanner

When any user logs in, a shell is started up. The shell has the terminal as standard input and standard output. It starts out by typing the prompt, a character such as a dollar sign, which tells the user that the shell is waiting to accept a command. If the user types

Date then the shell creates a child process and runs the date program as the child. When the child finishes, the shell types the prompt again and tries to read the next input line. The user can specify that standard output be redirected to a file, for example,

date >file

Similarly, standard input can be redirected, as in

sort <file1 >file2

This invokes the sort program with input taken from file1 and output sent to file2.

**Trap:** A trap is a software generated interrupt caused either by an error (for example, division by zero or invalid memory access) or by a specific request from a user that an operating system service be provided.

**System Calls**

The interface between a process and an operating system is provided by system calls. In general, system calls are available as assembly language instructions. They are also included in the manuals used by the assembly level programmers. System calls are usually made when a process in user mode requires access to a resource. Then it requests the kernel to provide the resource via a system call. A figure representing the execution of the system call is given as follows:

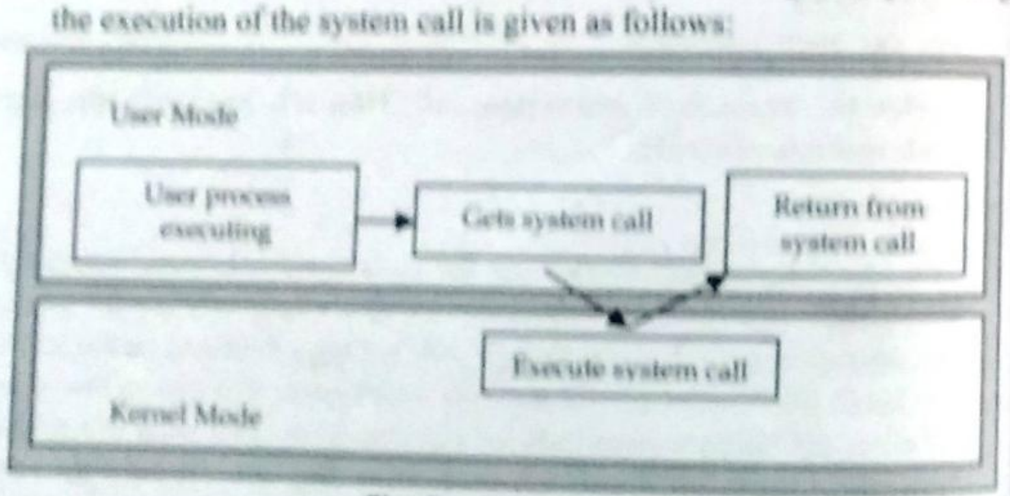



**Fig: System calls**

As can be seen from this diagram, the processes execute normally in the user mode until a system call interrupts this. Then the system call is executed on a priority basis in the kernel mode. After the execution of



the system call, the control returns to the user mode and execution of user processes can be resumed. In general, system calls are required in the following situations:

- If a file system requires the creation or deletion of files. Reading and writing from files also require a system call.
- Creation and management of new processes.
- Network connections also require system calls. This includes sending and receiving packets.
- Access to a hardware device such as a printer, scanner etc. requires a system call.

**5. What are main goals of interactive system scheduling? Discuss priority scheduling along with its pros and cons.**

**Solution:**

Computer scheduling where the process is either automatic or manually interrupted to allow the scheduler the opportunity to review and change the schedule is called interactive system scheduling.

**Priority based scheduling algorithm**

This is the scheduling algorithm which is based on the priority of the processes. In priority scheduling, the scheduler itself chooses the priority of the task, and then they will be executed according to their preference which is assigned to them by the scheduler. The process which has the highest priority will be firstly processed by the CPU, and the process which has the lowest priority will be executed by the CPU when the current process terminates its execution.

The priority scheduling algorithm supports preemptive and non - preemptive scheduling policy. The CPU is allocated to the process with the highest priority (smallest integer = highest priority). SJF is a priority scheduling where priority is the predicted next CPU burst time

**Example:** Consider the processes P1, P2, P3, P4 given in the below table, arrives for execution in the same order, with Arrival Time 0, and given Burst Time, let's find the average waiting time using the **Priority scheduling** algorithm.

| Process | Burst Time | Priority |
|---|---|---|
| P1 | 21 | 2 |
| P2 | 3 | 1 |
| P3 | 6 | 4 |
| P4 | 2 | 3 |

Scanned by TapScanner

The Gantt chart for following processes based on Priority scheduling will be

| P2 | P1 | P4 | P3 |
|---|---|---|---|

0 3 24 26 32

The average waiting time will be (0 + 3 + 24 + 26) / 4 = 13.25 ms

**Advantages of Priority Scheduling**

- The priority of process is selected on the basis of memory requirement, user preference or the requirement of time.
- Processes are executed on the basis of priority. So high priority does not need to wait for long which saves time
- It is easy to use.
- It is a user friendly algorithm.
- Simple to understand.
- It has reasonable support for priority.

**Disadvantages of Priority Scheduling**

- The major disadvantage of priority scheduling is the process of indefinite blocking or starvation. This problem appears when a process is ready to be executed but it has to wait for the long time for execution by CPU because other high priority processes are executed by the CPU. The problem of starvation can be solved by aging. Aging is a technique in which the system gradually increases the priority of those processes which are waiting in the system from a long time for their execution.
- In case if we have the processes which have the same priority, then we have to make use of 2nd scheduling algorithm.
- If the system gets crashes eventually, then all the processes having low priority which are not finished yet, also get lost.

**6. How starvation differs from deadlock? Consider the following situation of processes and resources:**

| Process | Has | Max |
|---|---|---|
| P1 | 2 | 6 |
| P2 | 1 | 5 |
| P3 | 2 | 5 |
| P4 | 2 | 6 |

Free=3

a. What will happen if process P3 requests 1 resource?
b. What will happen if process P4 requests 1 resource?



**Solution:**

**Difference between Starvation and Deadlock**

| S.N. | Deadlock | Starvation |
|---|---|---|
| 1 | Deadlock is a situation where no process got blocked and no process proceeds | Starvation is a situation where the low priority process got blocked and the high priority processes proceed. |
| 2 | Deadlock is an infinite waiting. | Starvation is a long waiting but not infinite. |
| 3 | Every Deadlock is always a starvation. | Every starvation need not be deadlock. |
| 4 | The requested resource is blocked by the other process. | The requested resource is continuously be used by the higher priority processes. |
| 5 | Deadlock happens when Mutual exclusion, hold and wait, No preemption and circular wait occurs simultaneously. | It occurs due to the uncontrolled priority and resource management. |

**7. Consider the segment table:**

**What are the physical address for the following logical addresses?**

a. 0,430
b. 1,10
c. 1,11
d. 2,500

| Segment | Base | Length |
|---|---|---|
| 0 | 219 | 600 |
| 1 | 2300 | 14 |
| 2 | 90 | 100 |
| 3 | 1327 | 580 |
| 4 | 1952 | 965 |

**Solution:**

a. 219+430 = 649.
b. 2300+10=2310.
c. 2300+11=2311
d. Illegal address since size of segment 2 is 100 and the offset in logicaladdress is 500.

**8. Define the term race condition? Justify that race condition leads data loss or incorrect data.**

**Solution:**

A race condition occurs when two or more threads can access shared data and they try to change it at the same time. Because the thread scheduling algorithm can swap between threads at any time, you don't

Scanned by TapScanner

know the order in which the threads will attempt to access the shared data. Therefore, the result of the change in data is dependent on the thread scheduling algorithm, i.e. both threads are "racing" to access/change the data.

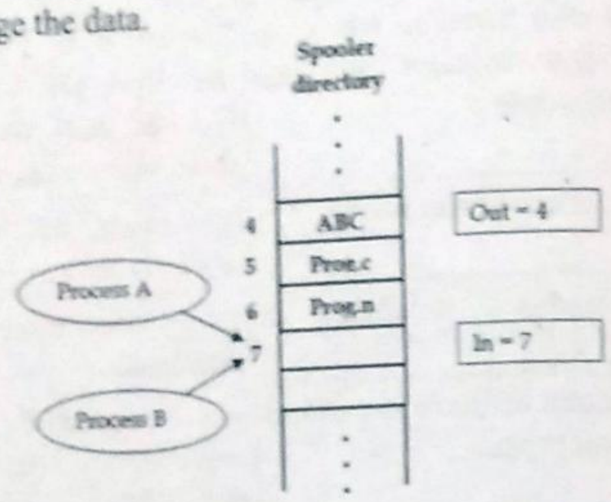



**Fig:** IPC with race condition

When a process wants to print a file, it enters the file name in a special spooler directory, another process, print daemon, periodically checks to see if there are any files to be printed, print them and removes their names from the directory.

Imagine that the spooler directory has a large number of slots, numbered 0, 1, 2, ..., each one capable of holding a file name. Also imagine that there are two shared variables, out, which points to the next file to be printed, and in, which points to the next free slot in the directory and these are available to all processes.

At a certain instant, slots 0 to 3 are empty and slots 4 to 6 are full. More or less simultaneously, process A and B decided to queue a file for printing. Process A reads in and stores the value 7 in a local variable called next_free_slot. Just then the CPU decides that process A has run long enough, so it switches to process B. Process B also reads in and also gets a 7 so stores the name of its file in slot 7 and updates in to be an 8. When process A runs again, starting from the place it left off, it finds a 7 in next_free_slot and writes its file name in slot 7, by erasing the name that process B just put there and then sets in to 8.

Situations like this, where two or more processes are reading or writing some shared data and the final result depends on who runs precisely when, are called race conditions.

**Solution to Race condition**

Make an operating system not to perform several tasks in parallel. This is called serialization.

Two strategies to serializing processes in a multitasking environment:

- The Scheduler can be disabled
- A Protocol can be introduced

Scheduler can be disabled for a short period of time, to prevent control being given to another process during a critical action like modifying



shared data. This will be inefficient on multiprocessor machines, since all other processors have to be halted every time one wishes to execute a critical section.

A protocol can be introduced which all programs sharing data must obey. The protocol ensures that processes have to queue up to gain access to shared data.

**9. Explain directory implementation techniques employed in operating systems briefly.**

**Solution:**

The selection of directory-allocation and directory-management algorithms significantly affects the efficiency, performance, and reliability of the file system. We implement the directory by using different two methods.

1. **Linear list:** The simplest method of implementing a directory is to use a linear list of file names with pointers to the data blocks. This method is simple to program but time-consuming to execute. To create a new file, we must first search the directory to be sure that no existing file has the same name. Then, we add a new entry at the end of the directory. To delete a file, we search the directory for the named file, then release the space allocated to it.The real disadvantage of a linear list of directory entries is that finding a file requires a linear search. Directory information is used frequently, and users will notice if access to it is slow. In fact, many operating systems implement a software cache to store the most recently used directory information. A cache hit avoids the need to constantly reread the information from disk. A sorted list allows a binary search and decreases the average search time.

**Characteristics**

- When a new file is created, then the entire list is checked whether the new file name is matching to an existing file name or not. In case, it doesn't exist, the file can be created at the beginning or at the end. Therefore, searching for a unique name is a big concern because traversing the whole list takes time.
- The list needs to be traversed in case of every operation (creation, deletion, updating, etc) on the files therefore the systems become inefficient.

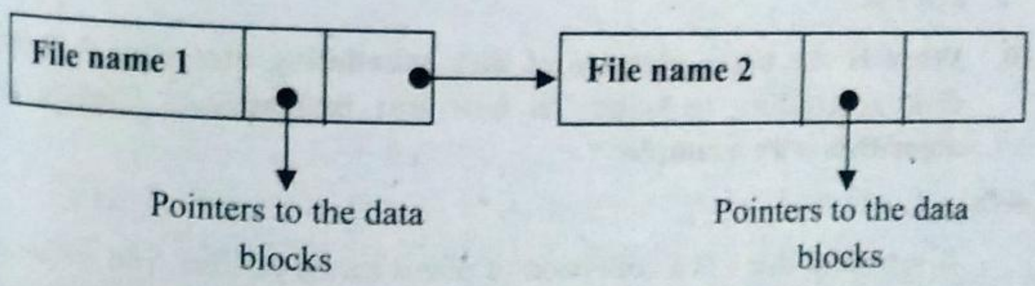
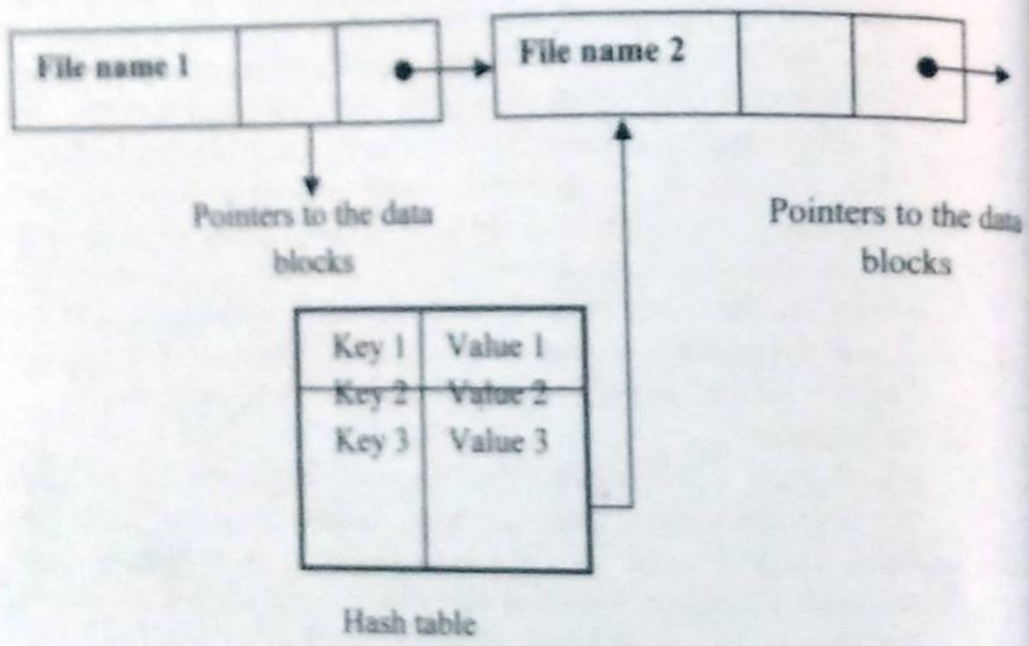



**Fig:** Linear List

Scanned by TapScanner

2. **Hash Table:** Another data structure that has been used for a file directory is a hash table. It consist linear list with hash table. The hash table takes a value computed from the file name and returns a pointer to the file name in the linear list. Therefore, it can greatly decrease the directory search time. Insertion and deletion are also fairly straightforward, although some provision must be made for **collisions**-situations where two file names hash to the same location. If that key is already in use, a linked list is constructed.

A key-value pair for each file in the directory gets generated and stored in the hash table. The key can be determined by applying the hash function on the file name while the key points to the corresponding file stored in the directory. Now, searching becomes efficient due to the fact that now, entire list will not be searched on every operating. Only hash table entries are checked using the key and if an entry found then the corresponding file will be fetched using the value.




**Advantages:** greatly decrease the file search time.

**Problem:** It greatly fixed size and dependence of the hash function on that size.

**10. What is the main purpose of disk scheduling algorithms? Which disk scheduling technique is best but impractical? Explain the algorithm with example.**

**Solution:**

A hard disk drive is a collection of plates called platters. The surface of each platter is divided into circular tracks. Furthermore, each track is divided into smaller pieces called sectors. Disk I/O is done sector by



sector. A group of tracks that are positioned on top of each other form a cylinder. There is a head connected to an arm for each surface, which handles all I/O operations. For each I/O request, first head is selected. It is then moved over the destination track. The disk is then rotated to position the desired sector under the head and finally, the read/write operation is performed. There are two objectives for any disk scheduling algorithm:

- **Maximize the throughput:** the average number of requests satisfied per time unit.
- **Minimize the response time:** the average time that a request must wait before it is satisfied.

Although there are other algorithms that reduce the seek time of all requests, we will only concentrate on the following major disk scheduling algorithms:

a. First Come-First Serve (FCFS)
b. Shortest Seek Time First (SSTF)
c. Elevator (SCAN)
d. Circular SCAN (C-SCAN)
e. LOOK
f. C-LOOK

SCAN and C-SCAN perform better for systems that place a heavy load on the disk.

**Elevator (SCAN)**

In SCAN algorithm the disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of disk, it reverses its direction and again services the request arriving in its path. So, this algorithm works like an elevator and hence also known as elevator algorithm. As a result, the requests at the midrange are serviced more and those arriving behind the disk arm will have to wait.

**Advantages**

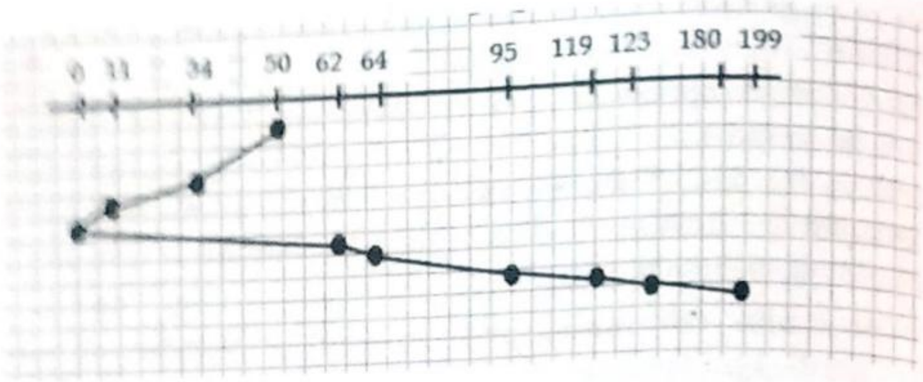
- High throughput
- Low variance of response time
- Average response time

**Disadvantages**

- Long waiting time for requests for locations just visited by disk arm

**Example:** Given the following queue {95, 180, 34, 119, 11, 123, 62, 64} with the Read-write head initially at the track 50 and the tail track being at 199. Then find total head movement by using SCAN disk scheduling algorithm.

Scanned by TapScanner




Total disk movements = (50-34) + (34-11) + (11-0) + (62-0) + (64-62) + (95-64)+(119-95)+(123-119)+(180-123)

= 16+23+11+62+2+31+24+4+57

= 230 tracks

**Circular SCAN (C-SCAN)**

In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area. These situations are avoided in CSAN algorithm in which the disk arm instead of reversing its direction goes to the other end of the disk and starts servicing the requests from there. So, the disk arm moves in a circular fashion and this algorithm is also similar to SCAN algorithm and hence it is known as Circular SCAN.

**Advantages**

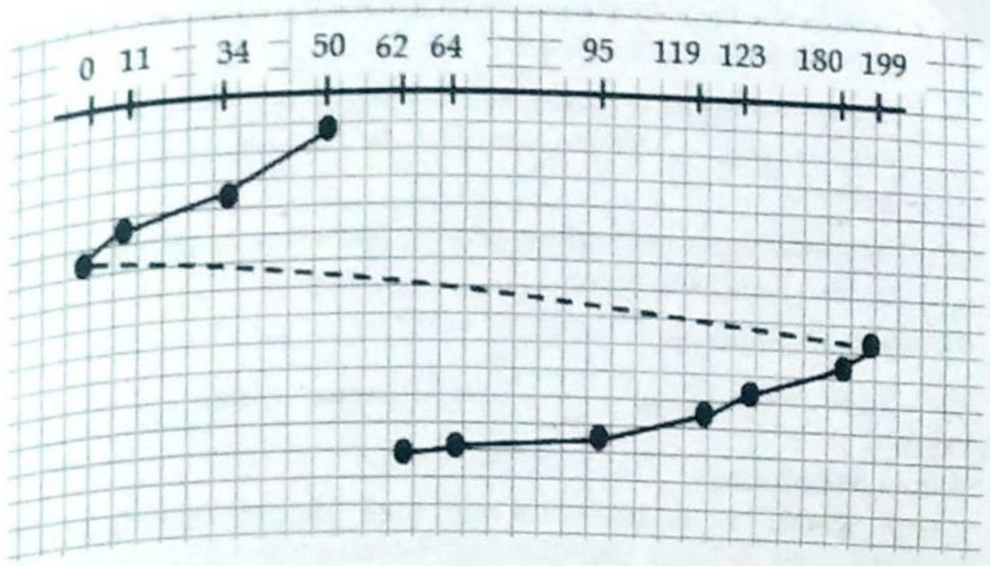
- The waiting time for the cylinders just visited by the head is reduced as compared to the SCAN Algorithm.
- It provides uniform waiting time.
- It provides better response time.

**Disadvantages**

- It causes more seek movements as compared to SCAN Algorithm.
- It causes the head to move till the end of the disk even if there are no requests to be serviced.

**Example:** Given the following queue {95, 180, 34, 119, 11, 123, 62, 64} with the Read-write head initially at the track 50 and the tail track being at 199. Then find total head movement by using CSCAN disk scheduling algorithm.






Total disk movements = (50-34)+(34-11)+(11-0)+(199-180)+(180-123)+(123-119)+(119-95)+(95-64)+(64-62)

= 16+23+11+19+57+4+24+31+2

= 187 tracks

**11. How threads differ from processes? Explain thread usages.**

**Solution:**

Some of the similarities and differences of thread and processes are:

**Similarities**

- Like processes threads share CPU and only one thread active (running) at a time.
- Like processes, threads within a process, threads within a process execute sequentially.
- Like processes, thread can create children.
- And like process, if one thread is blocked, another thread can run.

**Differences**

- All threads of a program are logically contained within a process.
- A process is heavy weighted, but a thread is light weighted.
- A program is an isolated execution unit whereas thread is not isolated and shares memory.
- A thread cannot have an individual existence; it is attached to a process. On the other hand, a process can exist individually.
- At the time of expiration of a thread, its associated stack could be recovered as every thread has its own stack. In contrast, if a process dies, all threads die including the process.

**Properties of a Thread**

- Only one system call can create more than one thread (Lightweight process).
- Threads share data and information.

Scanned by TapScanner

- Threads shares instruction, global and heap regions but have its own individual stack and registers.
- Thread management consumes no or fewer system calls as the communication between threads can be achieved using shared memory.
- The isolation property of the process increases its overhead in terms of resource consumption.

**12. Write short notes on:**

**a) Linux Scheduling** **b) Fragmentation**

**Solution:**

**Linux Scheduling**

Linux scheduling is based on the time sharing technique: several processes run in "time multiplexing" because the CPU time is divided into slices, one for each runnable process. If a currently running process is not terminated when its time slice or quantum expires, a process switch may take place. Time sharing relies on timer interrupts and is thus transparent to processes. No additional code needs to be inserted in the programs to ensure CPU time sharing. The scheduler always succeeds in finding a process to be executed; in fact, there is always at least one runnable process: the swapper process, which has PID 0 and executes only when the CPU cannot execute other processes.

Every Linux process is always scheduled according to one of the following scheduling classes:

**SCHED_FIFO**

A First-In, First-Out real-time process. When the scheduler assigns the CPU to the process, it leaves the process descriptor in its current position in the run queue list. If no other higher-priority real-time process is runnable, the process continues to use the CPU as long as it wishes, even if other real-time processes that have the same priority are runnable.

**SCHED_RR**

A Round Robin real-time process. When the scheduler assigns the CPU to the process, it puts the process descriptor at the end of the run queue list. This policy ensures a fair assignment of CPU time to all SCHED_RR real-time processes that have the same priority.

**SCHED_NORMAL**

It is a conventional, time-shared process.

**Fragmentation**

As processes are loaded and removed from memory, the free memory space is broken into littlepieces. It happens after sometimes that processes cannot be allocated to memory blocksconsidering their small size and memory blocks remains unused. This problem is known asFragmentation. Fragmentation is a condition that occurs when we dynamically allocate the RAM to the processes, then many free memory blocks are available but they are not enough to load the process on RAM. There are two types of fragmentations:



- Internal fragmentation and
- External fragmentation

**Internal fragmentation**

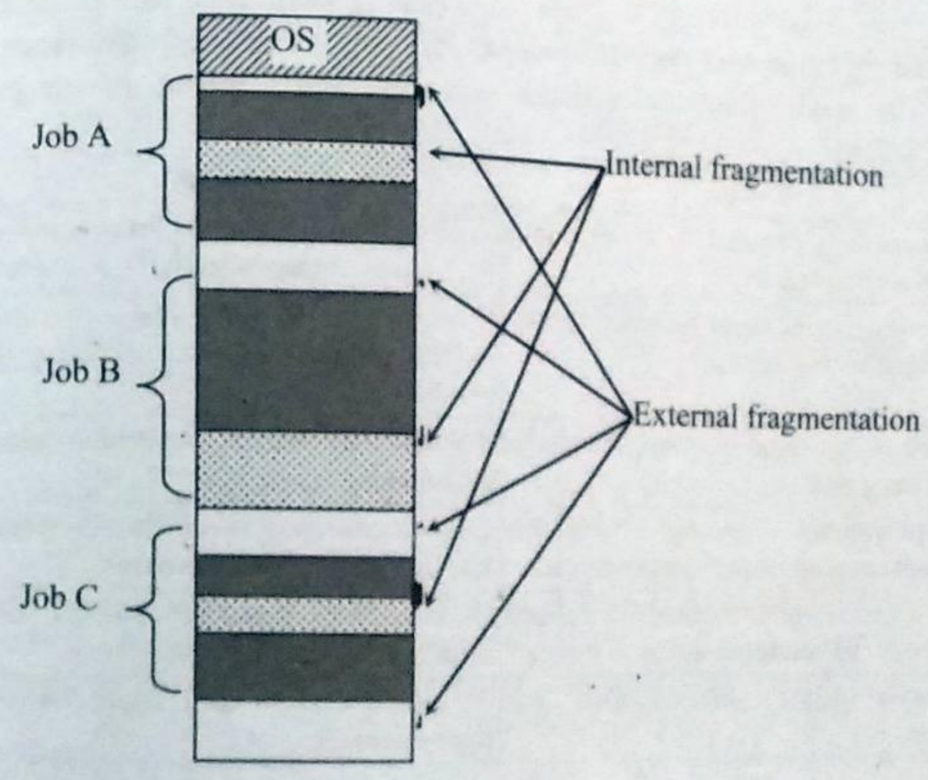
Internal fragmentation occurs when fixed sized memory blocks are available and a process gets a block that is too much larger than the storage requirement of process. Internal fragmentation occurs when the memory is divided into fixed sized blocks. Whenever a process request for the memory, the fixed sized block is allocated to the process. In case the memory assigned to the process is somewhat larger than the memory requested, then the difference between assigned and requested memory is the internal fragmentation.

**External fragmentation**

Total free RAM space is enough to load a process but the process still can't load because free blocks of RAM are not contiguous. In other words, we can say that all free blocks are not located together.

**Characteristics of external fragmentation**

- It exists when there is enough total memory space available to satisfy a request, but available memory space is not contiguous.
- Storage space is fragmented into large number of small holes.
- Both first fit and best fit strategies suffer from this.
- First fit is better in some systems, whereas best fit is better for other.
- Depending on the total amount of memory storage, size, external fragmentation may be minor or major problem.




Scanned by TapScanner

# MODEL QUESTIONSN-ANSWERS SET II

CSIT Second Year/ Fourth Semester — Full Marks: 60
Operating System [CSC259] — Pass Marks: 24
Time: 3 Hrs

***Candidates are required to give their answers in their own words as far as practicable.***

The figures in the margin indicate full marks.

## Group 'A'

**Long Answer Questions**

**Attempt any two questions.** **[2 × 10 = 20]**

**1. What resources are used when a thread is created? How do they differ from those used when a process is created?**

**Solution:**

Because a thread is smaller than a process, thread creation typically uses fewer resources than process creation. Creating a process requires allocating a process control block (PCB), a rather large data structure. The PCB includes a memory map, list of open files, and environment variables. Allocating and managing the memory map is typically the most time-consuming activity. Creating either a user or kernel thread involves allocating a small data structure to hold a register set, stack, and priority.

The primary difference is that threads within the same process run in a shared memory space, while processes run in separate memory spaces. Threads are not independent of one another like processes are, and as a result threads share with other threads their code section, data section, and OS resources (like open files and signals). But, like process, a thread has its own program counter (PC), register set, and stack space.

The major differences between a process and a thread are given as follows:

| Process | Thread |
|---|---|
| A process is a program under execution i.e. an active program. | A thread is a lightweight process that can be managed independently by a scheduler. |
| Processes require more time for context switching as they are heavier. | Threads require less time for context switching as they are lighter than processes. |
| Processes are totally independent and don't share memory. | A thread may share some memory with its peer threads. |
| Communication between processes requires more time than between threads. | Communication between threads requires less time than between processes. |
| If a process gets blocked, remaining processes can continue execution. | If a user level thread gets blocked, all of its peer threads also get blocked. |
| Processes require more resources than threads. | Threads generally need fewer resources than processes. |



| | |
|---|---|
| Individual processes are independent of each other. | Threads are parts of a process and so are dependent. |
| Processes have independent data and code segments. | A thread shares the data segment, code segment, files etc. with its peer threads. |
| All the different processes are treated separately by the operating system. | All user level peer threads are treated as a single task by the operating system. |
| Processes require more time for creation. | Threads require less time for creation. |
| Processes require more time for termination. | Threads require less time for termination. |

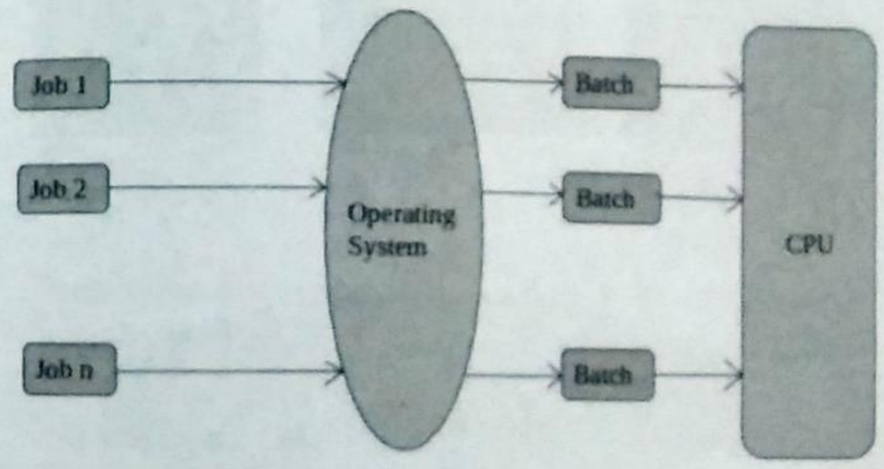
**2. What are the different types of operating systems? Explain them in detail.**

**Solution:**

An Operating System performs all the basic tasks like managing file, process, and memory. Thus operating system acts as manager of all the resources, i.e. resource manager. Thus operating system becomes an interface between user and machine. Types of Operating Systems: Some of the widely used operating systems are as follows:

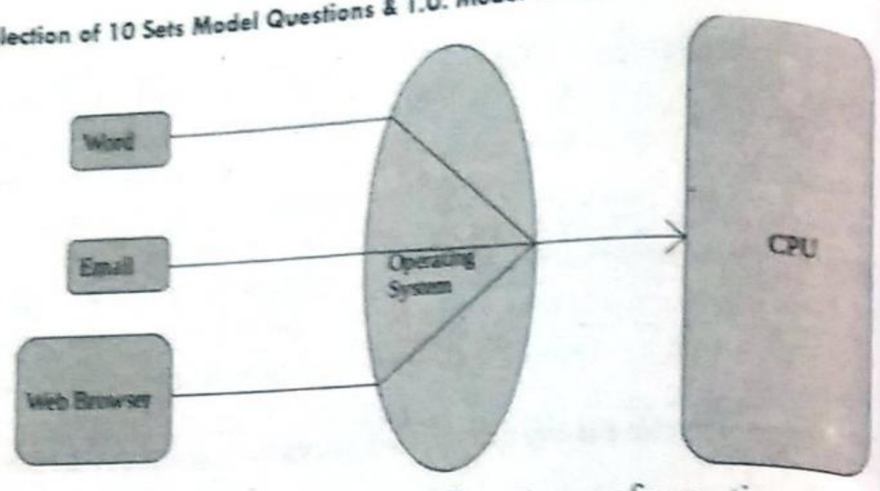
a. **Batch Operating System:** This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having same requirement and group them into batches. It is the responsibility of operator to sort the jobs with similar needs.

Examples of Batch based Operating System: Payroll System, Bank Statements etc.

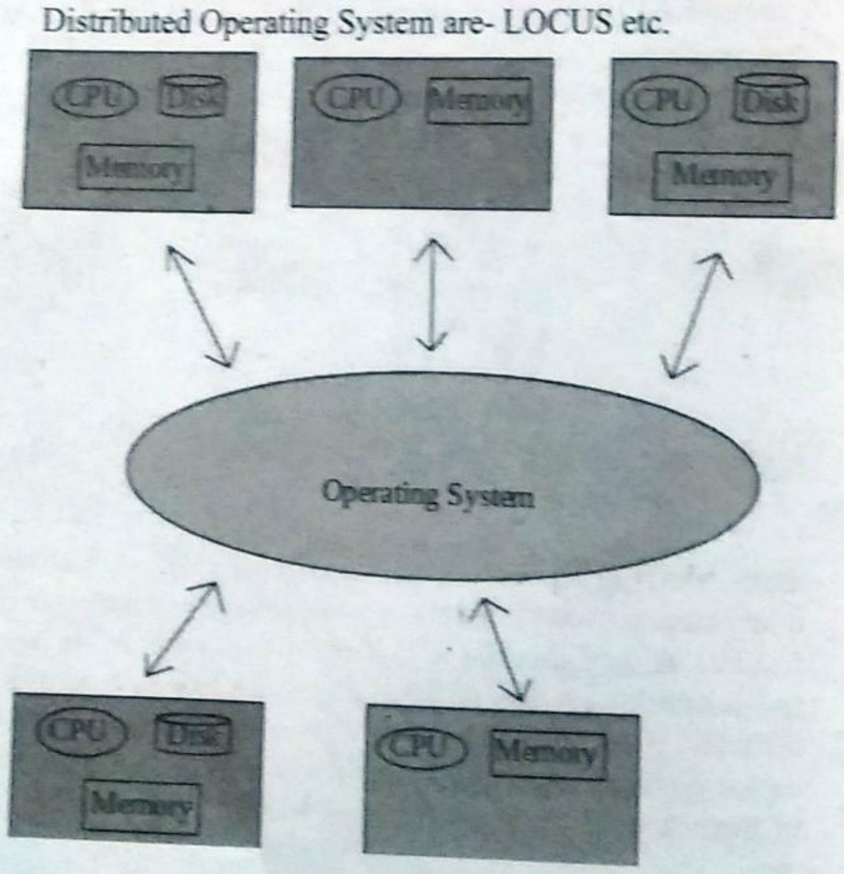



b. **Time-Sharing Operating Systems:** Each task is given some time to execute, so that all the tasks work smoothly. Each user gets time of CPU as they use single system. These systems are also known as Multitasking Systems. The task can be from single user or from different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to next task. Examples of Time-Sharing OSs are: Multics, Unix etc.

Scanned by TapScanner

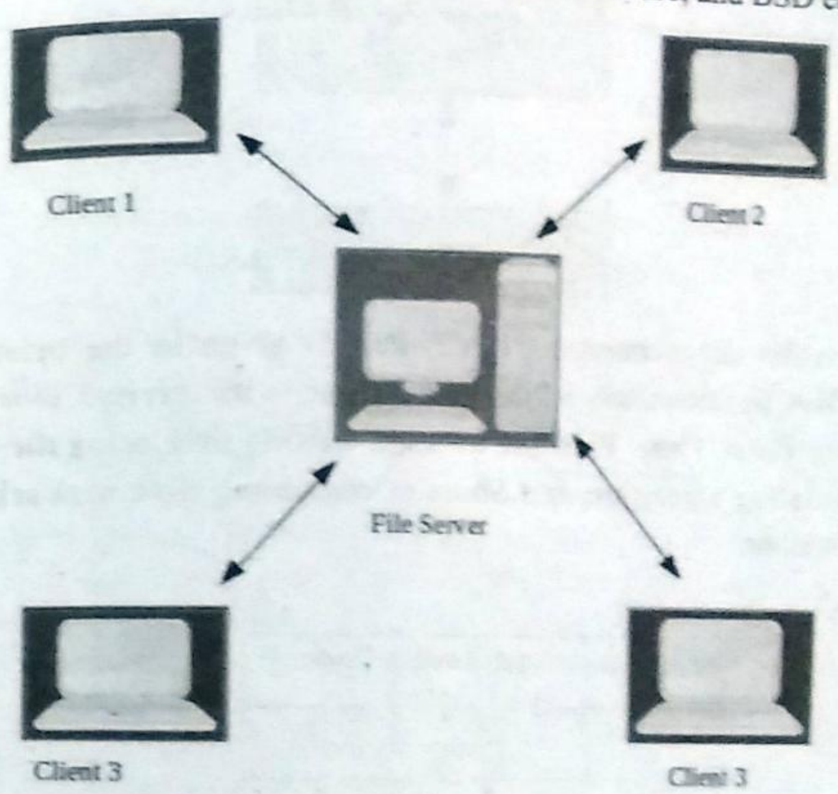



c. **Distributed Operating System:** These types of operating system is a recent advancement in the world of computer technology and are being widely accepted all-over the world and, that too, with a great pace. Various autonomous interconnected computers communicate each other using a shared communication network. Independent systems possess their own memory unit and CPU. These are referred as loosely coupled systems or distributed systems. These system's processors differ in size and function. The major benefit of working with these types of operating system is that it is always possible that one user can access the files or software which are not actually present on his system but on some other system connected within this network i.e., remote access is enabled within the devices connected in that network. Examples of Distributed Operating System are- LOCUS etc.






d. **Network Operating System:** These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions. These type of operating systems allow shared access of files, printers, security, applications, and other networking functions over a small private network. One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections etc. and that's why these computers are popularly known as tightly coupled systems. Examples of Network Operating System are: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD etc.



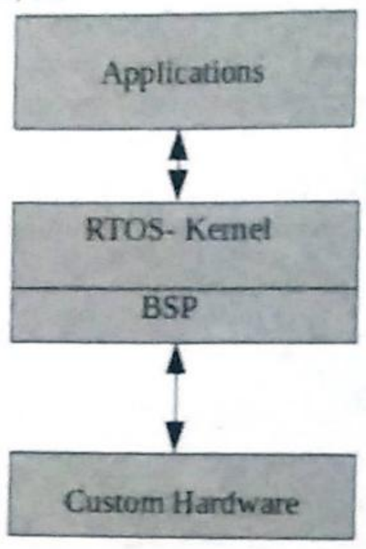

e. **Real-Time Operating System:** These types of OSs serves the real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called response time. Real-time systems are used when there are time requirements are very strict like missile systems, air traffic control systems, robots etc. Two types of Real-Time Operating System which are as follows:

**Hard Real-Time Systems:** These OSs are meant for the applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or air bags which are

Scanned by TapScanner

required to be readily available in case of any accident. Virtual memory is almost never found in these systems.

**Soft Real-Time Systems:** These OSs are for applications where for time-constraint is less strict. Examples of Real-Time Operating Systems are: Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.




**3. Consider the processes P1, P2, P3, P4 given in the below table, arrives for execution in the same order, with Arrival Time 0, and given Burst Time. Find the average waiting time using the Priority scheduling algorithm and Shortest remaining time next scheduling algorithm.**

**Solution:**

| Process | Burst Time | Priority |
|---|---|---|
| P1 | 21 | 2 |
| P2 | 3 | 1 |
| P3 | 6 | 4 |
| P4 | 2 | 3 |

The Gantt chart for following processes based on Priority scheduling will be

| P2 | P1 | P4 | P3 |
|---|---|---|---|

0 3 24 26 32

The average waiting time will be (0 + 3 + 24 + 26) / 4 = 13.25 ms



**Shortest remaining time next scheduling algorithm**

| Process | Burst Time | Arrival Time |
|---|---|---|
| P1 | 21 | 0 |
| P2 | 3 | 1 |
| P3 | 6 | 2 |
| P4 | 2 | 3 |

The Gantt chart for Preemptive Shortest remaining time next Scheduling will be

| | P1 | P2 | P4 | P2 | P3 | P1 |
|---|---|---|---|---|---|---|

0 1 3 5 6 1 32

The average waiting time will be ((5-3) + (6-2) + ( 12-1))/4 =4.25 ms

**Short Questions**

**Attempt any eight questions.** **(8 × 5 = 40)**

**4. Differentiate internal fragmentation and external fragmentation.**

**Solution:**

| Basis for comparison | Internal fragmentation | External fragmentation |
|---|---|---|
| Basic | It occurs when fixed sized memory blocks are allocated to the processes. | It occurs when variable size memory spaces are allocated to the processes dynamically. |
| Occurrence | When the memory assigned to the process is slightly larger than the memory requested by the process this creates free space in the allocated block causing internal fragmentation. | When the process is removed from the memory, it creates the free space in the memory causing external fragmentation. |
| Solution | The memory must be partitioned into variable sized blocks and assign the best fit block to the process. | Compaction, paging and segmentation. |

**5. What is Belady's anomaly? Explain with one example**

**Solution:**

In computer storage, Belady's anomaly is the phenomenon in which increasing the number of page frames results in an increase in the number of page faults for certain memory access patterns. This phenomenon is commonly experienced when using the first-in first-out (FIFO) page replacement algorithm.

Scanned by TapScanner

**For example**, if we consider reference string 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 and number of frames allocated=3, we get 9 total page faults, but if we increase slots to 4, we get 10 page faults.

**Reference string**

| 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 |
| | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 |
| | | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 4 | 4 |
| Pf | Pf | Pf | Pf | Pf | Pf | Pf | X | X | Pf | Pf | X |

Total page faults = 9

But if we use page frame size 4 for the same reference string we get 10 page faults as below;

| 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 4 | 4 |
| | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 5 |
| | | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 |
| | | | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 3 |
| Pf | Pf | Pf | Pf | X | X | Pf | Pf | Pf | Pf | Pf | Pf |

Total number of page faults = 10

6. **Consider the following page reference strings:**
**7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 How many page faults would occur for the following replacement algorithm, assuming three, four frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each.**
   a. **LRU replacement**
   b. **Optimal replacement**

**Optimal page replacement**

**Reference string**

| 7 | 0 | 1 | 2 | 0 | 3 | 0 | 4 | 2 | 3 | 0 | 3 | 2 | 1 | 2 | 0 | 1 | 7 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 7 | 7 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 7 | 7 | 7 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Pf | Pf | Pf | Pf | X | Pf | X | Pf | X | X | Pf | X | X | Pf | X | X | X | Pf | X | X |

Total page faults = 9

LRU page replacement

**Reference string**

| 7 | 0 | 1 | 2 | 0 | 3 | 0 | 4 | 2 | 3 | 0 | 3 | 2 | 1 | 2 | 0 | 1 | 7 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 7 | 7 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| | | 1 | 1 | 1 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 7 | 7 | 7 |
| Pf | Pf | Pf | Pf | X | Pf | X | Pf | Pf | Pf | Pf | X | X | Pf | X | Pf | X | Pf | X | X |

Total page faults = 12



7. **List the conditions that must be prevention for deadlock to occur and for each condition give brief example or reason that illustrate a disadvantage in preventing the condition.**

**Solution:**

If we simulate deadlock with a table which is standing on its four legs then we can also simulate four legs with the four conditions which when occurs simultaneously, cause the deadlock.

However, if we break one of the legs of the table then the table will fall definitely. The same happens with deadlock, if we can be able to violate one of the four necessary conditions and don't let them occur together then we can prevent the deadlock. Let's see how we can prevent each of the conditions.

a. **Mutual Exclusion:** Mutual section from the resource point of view is the fact that a resource can never be used by more than one process simultaneously which is fair enough but that is the main reason behind the deadlock. If a resource could have been used by more than one process at the same time then the process would have never been waiting for any resource. However, if we can be able to violate resources behaving in the mutually exclusive manner then the deadlock can be prevented.

**Spooling**

For a device like printer, spooling can work. There is a memory associated with the printer which stores jobs from each of the process into it. Later, Printer collects all the jobs and print each one of them according to FCFS. By using this mechanism, the process doesn't have to wait for the printer and it can continue whatever it was doing.

Although, Spooling can be an effective approach to violate mutual exclusion but it suffers from two kinds of problems.

- This cannot be applied to every resource.
- After some point of time, there may arise a race condition between the processes to get space in that spool.

We cannot force a resource to be used by more than one process at the same time since it will not be fair enough and some serious problems may arise in the performance. Therefore, we cannot violate mutual exclusion for a process practically.

b. **Hold and Wait:** Hold and wait condition lies when a process holds a resource and waiting for some other resource to complete its task. Deadlock occurs because there can be more than one process which are holding one resource and waiting for other in the cyclic order. However, we have to find out some mechanism by which a process either doesn't hold any resource or doesn't wait. That means, a

Scanned by TapScanner

process must be assigned all the necessary resources before the execution starts. A process must not wait for any resource once the execution has been started. This can be implemented practically if a process declares all the resources initially. However, this sounds very practical but can't be done in the computer system because a process can't determine necessary resources initially.

Process is the set of instructions which are executed by the CPU. Each of the instruction may demand multiple resources at the multiple times. The need cannot be fixed by the OS.

The problem with the approach is:

- Practically not possible.
- Possibility of getting starved will be increases due to the fact that some process may hold a resource for a very long time.

c. **No Preemption:** Deadlock arises due to the fact that a process can't be stopped once it starts. However, if we take the resource away from the process which is causing deadlock then we can prevent deadlock. This is not a good approach at all since if we take a resource away which is being used by the process then all the work which it has done till now can become inconsistent. Consider a printer is being used by any process. If we take the printer away from that process and assign it to some other process then all the data which has been printed can become inconsistent and ineffective and also the fact that the process can't start printing again from where it has left which causes performance inefficiency.

d. **Circular Wait:** To violate circular wait, we can assign a priority number to each of the resource. A process can't request for a lesser priority resource. This ensures that not a single process can request a resource which is being utilized by some other process and no cycle will be formed.

**8. What is a monitor? Explain how dinning philosopher's problem is solved using monitors with example pseudo code.**

**Solution:**

Monitors are a synchronization construct that were created to overcome the problems caused by semaphores such as timing errors. Monitors are abstract data types and contain shared data variables and procedures. The shared data variables cannot be directly accessed by a process and procedures are required to allow a single process to access the shared data variables at a time. This is demonstrated as follows:



```
Monitor monitor Name
{
    data variables;
    Condition variables;
    Procedure P1 {.....}
    Procedure P2 {.....}
    Procedure Pn {.....}
}
```

**Condition Variables**

Two different operations are performed on the condition variables of the monitor.

Wait.

Signal.

Let say we have 2 condition variables

Condition x, y; //Declaring variable

**Wait operation**

**x.wait():** Process performing wait operation on any condition variable are suspended. The suspended processes are placed in block queue of that condition variable.

**Note:** Each condition variable has its unique block queue.

**Signal operation**

x.signal(): When a process performs signal operation on condition variable, one of the blocked processes is given chance.

```
If (x block queue empty)
        // ignore signal
else
        // Resume a process from block queue.
```

Only one process can be active in a monitor at a time. Other processes that need to access the shared variables in a monitor have to line up in a queue and are only provided access when the previous processes release the shared variables.

**Dining Philosopher Problem**

Dijkstra posed and solved a synchronization problem he called the dining philosophers problem. Since that time, everyone inventing yet another synchronization primitive has felt obligated to demonstrate how wonderful the new primitive is by showing how elegantly it solves the dining philosophers' problem.

Scanned by TapScanner

**Fig:** Dining philosopher problem

There are N philosophers sitting around a circular table eating spaghetti and discussing philosophy. The problem is that each philosopher needs 2 forks to eat, and there are only N forks, one between each 2 philosophers. Design an algorithm that the philosophers can follow that insures that none starves as long as each philosopher eventually stops eating, and such that the maximum numbers of philosopher scan eat at once.

- Philosophers eat/think
- Eating needs 2 forks
- Pick one fork at a time
- How to prevent deadlock

The problem was designed to illustrate the problem of avoiding deadlock, a system state in which no progress is possible. One idea is to instruct each philosopher to behave as follows:

- think until the left fork is available; when it is, pick it up
- think until the right fork is available; when it is, pick it up
- eat
- put the left fork down
- put the right fork down
- repeat from the start

The solution presented below is deadlock-free and allows the maximum parallelism for an arbitrary number of philosophers. It uses an array, state, to keep track of whether a philosopher is eating, thinking, or hungry (trying to acquire forks). A philosopher may move into eating state only if neither neighbor is eating. Philosopher is neighbors are defined by the macros LEFT and RIGHT. In other words, if i is 2, LEFT is 1 and RIGHT is 3.



```
#define N 5                          /* number of philosophers */
#define LEFT (i+N-1)%N               /* number of i's left neighbor */
#define RIGHT (i+1)%N                /* number of i's right neighbor */
#define THINKING 0                   /* philosopher is thinking */
#define HUNGRY 1                     /* philosopher is trying to get forks */
#define EATING 2                     /* philosopher is eating */
Typedef int semaphore;               /* semaphores are a special kind of int */
int state[N];                        /* array to keep track of everyone's state */
semaphore mutex = 1;                 /* mutual exclusion for critical regions */
semaphore s[N];                      /* one semaphore per philosopher */
void philosopher (int i)             /* i: philosopher number, from 0 to N-1 */
{
        while (TRUE)
        {                            /* repeat forever */
                think();             /* philosopher is thinking */
                take_forks(i);       /* acquire two forks or block */
                eat();               /* yum-yum, spaghetti */
                put_forks(i);        /* put both forks back on table */
        }
}
voidtake_forks(int i)                /* i: philosopher number, from 0 to N-1 */
{
        down(&mutex);                /* enter critical region */
        state[i] = HUNGRY;           /* record fact that philosopher i is hungry */
        test(i);                     /* try to acquire 2 forks */
        up(&mutex);                  /* exit critical region */
        down(&s[i]);                 /* block if forks were not acquired */
}
voidput_forks(i)                     /* i: philosopher number, from 0 to N-1 */
{
        down(&mutex);                /* enter critical region */
        state[i] = THINKING;         /* philosopher has finished eating */
        test(LEFT);                  /* see if left neighbor can now eat */
        test(RIGHT);                 /* see if right neighbor can now eat */
        up(&mutex);                  /* exit critical region */
}
void test(i)                         /* i: philosopher number, from 0 to N-1 */
{
if(state[i] == HUNGRY && state[LEFT] != EATING && state[RIGHT] != EATING)
        {
                state[i] = EATING;
                up(&s[i]);
        }
}
```

**9. Consider a disk with a mean seek time of 8ms, a rotational rate of 15,000 rpm, and 262,144 bytes per track. What are the access times for block sizes of 1 KB, 2 KB, and 4 KB, respectively?**

**Solution:** Given,

Average seek time, $T_s = 8ms = 8 \times 10^{-3}$ s

Rotational rate, $r = 15000 \text{ rpm} = \frac{15000}{60} \text{ rps} = 250 \text{ rps}$

Scanned by TapScanner

Number of bytes per track, N = 262144

Hence, access time for 1 KB block = $Ts + \frac{1}{2r} + \frac{1024}{r \times N}$

$= 8 \times 10^{-3} + \frac{1}{2 \times 250} + \frac{1024}{250 \times 26214} = 10.0156$ ms

Hence, access time for 2 KB block = $Ts + \frac{1}{2r} + \frac{1024}{r \times N} + \frac{1}{2r} + \frac{1024}{2 \times N}$

$= 8 \times 10^{-3} + \frac{1}{2 \times 250} + \frac{1024 \times 2}{250 \times 262144} = 10.03125$ ms

Hence, access time for 4 KB block = $Ts + \frac{1}{2r} + \frac{1024}{r \times N}$

$= 8 \times 10^{-3} + \frac{1}{2 \times 250} + \frac{1024 \times 4}{250 \times 262144} = 10.0625$ ms

**10. Explain the following disk scheduling algorithm with proper diagram**

**a) FCFS** **b) SSTF**
**c) SCAN** **d) LOOK**
**e) C-SCAN**

**First Come-First Serve (FCFS)**

FCFS is the simplest of all the Disk Scheduling Algorithms. In FCFS, the requests are addressed in the order they arrive in the disk queue. Since no reordering of request takes place the head may move almost randomly across the surface of the disk. This policy aims to minimize response time with little regard for throughput.

**Features of FCFS**

- perform operations in order requested
- no reordering of work queue
- **no starvation:** every request is serviced
- poor performance

**Advantages**

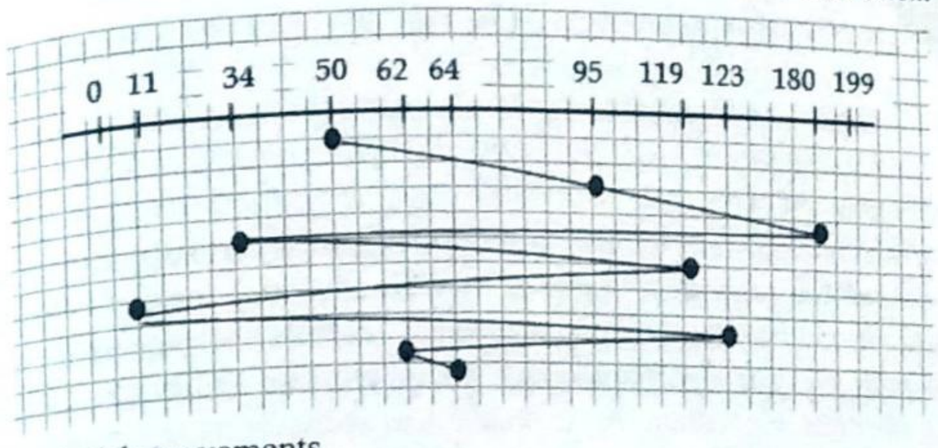
- Every request gets a fair chance
- No indefinite postponement

**Disadvantages**

- Does not try to optimize seek time
- May not provide the best possible service

**Example:** Given the following queue {95, 180, 34, 119, 11, 123, 62, 64} with the Read-write head initially at the track 50 and the tail track being at 199. Then find total head movement by using FCFS disk scheduling algorithm.






Total disk movements

= (95-50)+(180-95)+(180-34)+(119-34)+(119-11)+(123-11) +(123-62)+(64-62) = 45+85+146+85+108+112+61+2 = 644 tracks

**Shortest Seek Time First (SSTF)**

In SSTF requests having shortest seek time are executed first. So, the seek time of every request is calculated in advance in queue and then they are scheduled according to their calculated seek time. As a result, the request near the disk arm will get executed first. SSTF is certainly an improvement over FCFS as it decreases the average response time and increases the throughput of system. This policy will have better throughput than FCFS but a request may be delayed for a long period if many closely located requests arrive just after it.

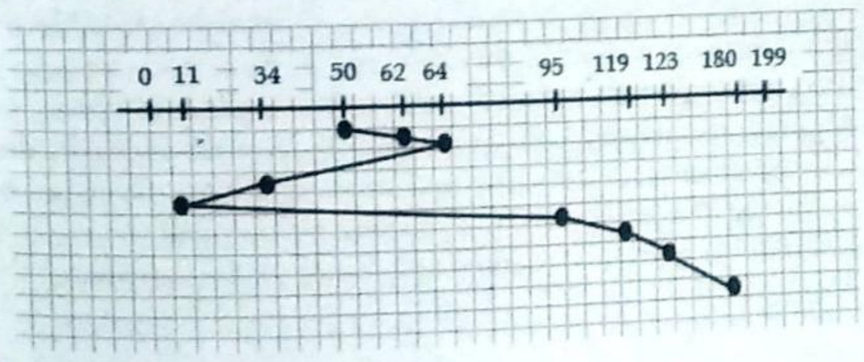
**Advantages**

- Average Response Time decreases
- Throughput increases

**Disadvantages**

- Overhead to calculate seek time in advance
- Can cause Starvation for a request if it has higher seek time as compared to incoming requests
- High variance of response time as SSTF favors only some requests

**Example:** Given the following queue {95, 180, 34, 119, 11, 123, 62, 64} with the Read-write head initially at the track 50 and the tail track being at 199. Then find total head movement by using SSTF disk scheduling algorithm.




Total disk movements

Scanned by TapScanner

= (62-50)+(64-62)+(64-34)+(34-11)+(95-11)+(119-95)+(123-119)+(180-123)
= 12+2+30+23+84+24+4+57
= 236 tracks

**Elevator (SCAN)**

In SCAN algorithm the disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of disk, it reverses its direction and again services the request arriving in its path. So, this algorithm works like an elevator and hence also known as elevator algorithm. As a result, the requests at the midrange are serviced more and those arriving behind the disk arm will have to wait.

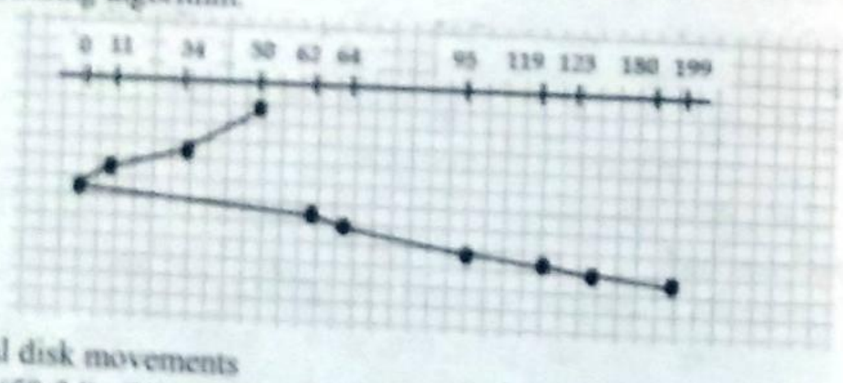
**Advantages**

- High throughput
- Low variance of response time
- Average response time

**Disadvantages**

- Long waiting time for requests for locations just visited by disk arm

**Example:** Given the following queue {95, 180, 34, 119, 11, 123, 62, 64} with the Read-write head initially at the track 50 and the tail track being at 199. Then find total head movement by using SCAN disk scheduling algorithm.




Total disk movements

= (50-34)+(34-11)+(11-0)+(62-0)+(64-62)+(95-64)+(119-95)+(123-119)+(180-123)
= 16+23+11+62+2+31+24+4+57
= 230 tracks

**Circular SCAN (C-SCAN)**

In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area. These situations are avoided in CSAN algorithm in which the disk arm instead of reversing its direction goes to the other end of the disk and starts servicing the requests from there. So, the disk arm moves in a circular fashion and this algorithm is also similar to SCAN algorithm and hence it is known as Circular SCAN.



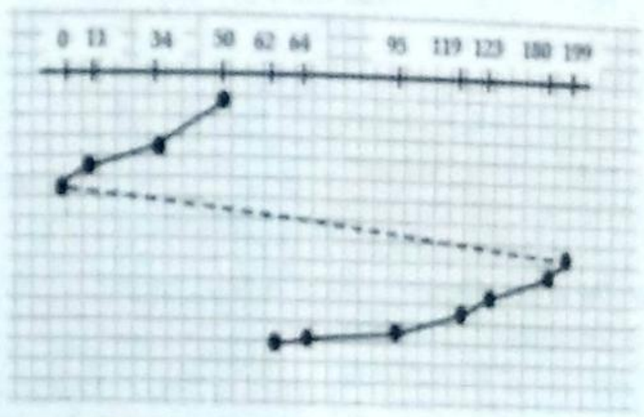
**Advantages**

- The waiting time for the cylinders just visited by the head is reduced as compared to the SCAN Algorithm.
- It provides uniform waiting time.
- It provides better response time.

**Disadvantages**

- It causes more seek movements as compared to SCAN Algorithm.
- It causes the head to move till the end of the disk even if there are no requests to be serviced.

**Example:** Given the following queue {95, 180, 34, 119, 11, 123, 62, 64} with the Read-write head initially at the track 50 and the tail track being at 199. Then find total head movement by using CSCAN disk scheduling algorithm.




Total disk movements

= (50-34)+(34-11)+(11-0)+(199-180)+(180-123)+(123-119)+(119-95)+(95-64)+(64-62)
= 16+23+11+19+57+4+24+31+2
= 187 tracks

**LOOK**

It is similar to the SCAN disk scheduling algorithm except the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be serviced in front of the head and then reverses its direction from there only. Thus it prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.

The main difference between SCAN Algorithm and LOOK Algorithm is:

- SCAN Algorithm scans all the cylinders of the disk starting from one end to the other end even if there are no requests at the ends.
- LOOK Algorithm scans all the cylinders of the disk starting from the first request at one end to the last request at the other end.

**Advantages**

- It does not cause the head to move till the ends of the disk when there are no requests to be serviced.
- It provides better performance as compared to SCAN Algorithm.
- It does not lead to starvation.

Scanned by TapScanner

- It provides low variance in response time and waiting time.

**Disadvantages**

- There is an overhead of finding the end requests.
- It causes long waiting time for the cylinders just visited by the head.

**11. Given memory partitions of 100K, 500K, 200K, 300K, and 600K (in order), how would each of the First-Fit, Best-Fit, and Worst-Fit algorithms place processes of 212K, 417K, 112K, and 426K (in order)? Which algorithm makes the most efficient use of memory?**

**Solution:**

**For First-Fit algorithm**

1. 212K is put in 500K partition
2. 417K is put in 600K partition
3. 112K is put in 288K partition (new partition 288K = 500K - 212K)
4. 426K must wait

**For Best-Fit algorithm**

1. 212K is put in 300K partition
2. 417K is put in 500K partition
3. 112K is put in 200K partition
4. 426K is put in 600K partition

**For Worst-Fit algorithm**

1. 212K is put in 600K partition
2. 417K is put in 500K partition
3. 112K is put in 388K partition
4. 426K must wait

In this example, Best-Fit turns out to be the best.

**12. What are the major activities of an operating system with regard to file management? Explain them briefly with their supporting system calls.**

**Solution:**

A file is a collected of related information defined by its creator. Computer can store files on the disk (secondary storage), which provide long term storage. Some examples of storage media are magnetic tape, magnetic disk and optical disk. Each of these media has its own properties like speed, capacity, data transfer rate and access methods.A file system normally organized into directories to ease their use. These directories may contain files and other directions.The five main major activities of an operating system in regard to file management are

- The creation and deletion of files.
- The creation and deletion of directions.
- The support of primitives for manipulating files and directions.
- The mapping of files onto secondary storage.
- The backup of files on stable storage media.



# Model Questions Sets For Practice

## MODEL SET 1

CSIT Second Year/ Fourth Semester
Operating System [CSC259]

Full Marks: 60
Pass Marks: 24
Time: 3 Hrs

***Candidates are required to give their answers in their own words as far as practicable.***

The figures in the margin indicate full marks.

**Section 'A'**

**Long Answer Questions**

**Attempt any two questions.** **(2 × 10=20)**

1. List the conditions that must be present for deadlock to occur and for each condition give brief example or reason that illustrate a disadvantage in preventing the condition.
2. Consider a swapping system in which memory consists of the following hole sizes in memory order: 10 KB, 4 KB, 20 KB, 18 KB, 7 KB, 9 KB, 12 KB, and 15 KB. Whichhole is taken for successive segment requests of: (i) 12 KB (ii) 10 KB (iii) 9 KB forfirst fit, best fit, worst fit, and nextfit approaches?
3. Consider the following snapshot of a system:

| Processes | Allocation | Max | Available |
|---|---|---|---|
| | A B C D | A B C D | A B C D |
| P0 | 0 0 1 2 | 0 0 1 2 | 2 1 0 0 |
| P1 | 2 0 0 0 | 2 7 5 0 | |
| P2 | 0 0 3 4 | 6 6 5 6 | |
| P3 | 2 3 4 5 | 4 3 5 6 | |
| P4 | 0 3 3 2 | 0 6 5 2 | |

Answer the following questions using the banker's algorithm:

a. What is the content of the matrix Need?

b. Is the system in a safe state? Why?

c. Is the system currently deadlocked? Why or why not?

d. Which process, if any, or may become deadlocked if this whole request is granted immediately?

Scanned by TapScanner

**Short Questions**

**Attempt any eight questions.** **(8 × 5 = 40)**

4. Describe dining-philosopher problem? Device an algorithm to solve the problem using semaphores?
5. Define process. How many different states a process has? Explain when a process changes the state with a state diagram?
6. Compare the performance of write operations achieved by a RAID level 5 organizationwith that achieved by a RAID level 0 organizations?
7. Consider the following set of processes with the length of the CPU burst time given in milliseconds

| Process | Burst Time | Priority |
|---|---|---|
| P1 | 10 | 3 |
| P2 | 1 | 1 |
| P3 | 2 | 3 |
| P4 | 1 | 4 |
| P5 | 5 | 2 |

The processes are assumed to have arrived in the order p1, p2, p3, p4, p5 all at time 0.

a. Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, anon pre-emptive priority (a smaller priority number implies a higher priority) and RR (quantum=1) scheduling.

b. What is the turnaround time of each process for each of the scheduling algorithms in part?

c. What is the waiting time of each process for each of the scheduling algorithms in part? Which of the schedules in part a results in the minimal average waiting time?

8. Describe resource-allocation graph? Explain how resource graph can beused for detecting deadlocks?
9. Consider the page reference string 1,3,4,0,5,3,2,1,0,4,5,2.How many page faults occur for the LRU and Optimal replacement algorithms with 4 frames each ?
10. What is thrashing? Explain the Causes of Thrashing.
11. Explain the concept of file sharing. What are the criteria to be followed in systems whichimplement file sharing?
12. What are the main differences between operating systems for mainframe computers and personal computers?



# MODEL SET 2

## Section 'A'

**Long Answer Questions**

**Attempt any two questions.** **(2 × 10=20)**

1. What is a monitor? Explain how dinning philosopher's problem is solved using monitors with example pseudo code.
2. Suppose that a disk drive has 5000 cylinders numbered 0 to 4999. The drive is currently serving a request at cylinder 143. The queue of pending requests in FIFO order86,1470,913,1774,948,1509, 1022, 1750, 130 starting from current head position. What is the total distance that disk arm moves to satisfy all the pending request for FCFS and SSTF disk scheduling algorithm?
3. Consider the following snapshot of a system

| | Allocation | | | | Max | | | | Available | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | 0 | 0 | 1 | 3 | 0 | 0 | 1 | 2 | 1 | 5 | 2 | 0 |
| P2 | 1 | 0 | 0 | 0 | 1 | 7 | 5 | 0 | | | | |
| P3 | 1 | 3 | 5 | 4 | 2 | 3 | 5 | 6 | | | | |

Answer the following questions using the banker's algorithm:

a. What is the content of matrix "Need"?

b. Is the system in a safe state?

c. If a request from process P1 arrives for (0, 4, 2, 0) can the request be granted immediately?

**Short Questions**

**Attempt any eight questions.** **(8 × 5 = 40)**

4. Define deadlock? What are the four conditions necessary for a deadlock situation to arise? How it can be prevented?
5. Consider a logical address space of eight pages of 1024 words each mapped onto a physical memory of 32 frames

   a. How many bits are in the logical address?

   b. How many bits are in the physical address?
6. Discuss the Peferson's solution for the race condition with algorithm.
7. Define Essential properties of Distributed Operating Systems.
8. What are the main differences between operating systems for mainframe computers and personal computers?
9. What is a Directory? Write short note on Directory implementation.

Scanned by TapScanner

10. Explain the terms in Memory Partitioning with examples:
    a. Fixed Partitioning
    b. Dynamic partitioning.
11. Suppose the head of a moving head disk with 200 tracks, numbered 0 to 199, is currently serving a request at track 143 and has just finished a request at track 125.If the queue of requests is kept in FIFO order: 86, 147, 91, 177, 94, 150, 102, 175, 130. What is the total head movement to satisfy these requests for the following disk scheduling algorithms?
    (a) FCFS (b) SCAN
    (b) (c) SSTF (d) C- SCAN
12. What is Belady's anomaly? Explain with one example

## MODEL SET 3

### Section "A"

**Long Answer Questions**

**Attempt any two questions.** (2 × 10=20)

1. Define deadlock? What are the four conditions necessary for a deadlocksituation to arise? How it can be prevented?
2. Mention the various file-directory structures. Brief about physical directoryimplementation.
3. Following is the snapshot of a CPU

| Process | CPU Burst | Arrival Time |
|---|---|---|
| P1 | 75 | 0 |
| P2 | 40 | 10 |
| P3 | 25 | 10 |
| P4 | 20 | 80 |
| P5 | 45 | 85 |

Draw the Gantt chart and calculate the turnaround time and waiting time of the jobs for
    a. FCFS (First Come First Served),
    b. SJF (Shortest Job First),
    c. SRTF (Shortest RemainingTime First) and
    d. RR (Round Robin with time quantum 15) scheduling algorithms

**Short Questions**

**Attempt any eight questions. (8 × 5 = 40)**

4. Define thrashing? Explain the different methods to avoid thrashing
5. Explain how double buffering improves the performance than a single buffer for I/O?



6. Consider the following page-reference string: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6 Calculate the number of page faults would occur for the following replacement algorithms, assuming frame size is 4. Remember that frames are initially empty.
    a. LRU replacement
    b. FIFO replacement
    c. Optimal replacement
7. Explain the conditions for the deadlock to occur? How can a deadlock be prevented?
8. Describe the differences between preemptive scheduler and non-preemptive scheduler.
9. Consider a swapping system in which memory consists of the following hole sizes in memory order: 10 KB, 4 KB, 20 KB, 18 KB, 7 KB, 9 KB, 12 KB, and 15 KB. Whichhole is taken for successive segment requests of: (i) 12 KB (ii) 10 KB (iii) 9 KB forfirst fit, best fit, worst fit, and nextfit approaches.
10. Describe dining-philosopher problem? Devise an algorithm to solve the problem usingsemaphores?
11. What are the major activities of an operating system with regard to file management? Explain them briefly with their supporting system calls.
12. Define the essential properties of Interactive operating systems and Network operating systems.

## MODEL SET 4

### Section "A"

**Long Answer Questions**

**Attempt any two questions.** (2 × 10=20)

1. List the conditions that must be present for deadlock to occur and for each condition give brief example or reason that illustrate a disadvantage in preventing the condition.
2. Consider the following set of process, with the length of the CPU burst given in milliseconds.

| Process | Burst Time | Priority |
|---|---|---|
| P1 | 10 | 3 |
| P2 | 1 | 1 |
| P3 | 2 | 3 |
| P4 | 1 | 4 |
| P5 | 5 | 2 |

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.What is the turnaround time of each process by applying priority scheduling algorithm?

Scanned by TapScanner

3. Discuss readers/writers problem and give solution by using semaphores where readershave priority.

**Short Questions**

**Attempt any eight questions.** **(8 × 5 = 40)**

4. State the purpose of short-term, medium-term and long term schedulers. Also discuss the differences among them.
5. What system calls have to be executed by a command interpreter or shell in order to start a new process? Explain briefly.
6. What are the different types of operating systems? Explain them in detail.
7. Construct the Gantt chart for
   a) Shortest job first
   b) Round Robin with q=3
   c) Roundrobin with q=4
   d) Shortest remaining time first scheduling algorithms for the following.

| Process | Arrival time | CPU burst Time in ms |
|---|---|---|
| P1 | 0 | 10 |
| P2 | 0 | 6 |
| P3 | 2 | 12 |
| P4 | 1 | 8 |
| P5 | 3 | 5 |

8. What resources are used when a thread is created? How do they differ from those used when a process is created?
9. Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder125. The queue of pending requests, in FIFO order, is: 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130. Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the SSTF disk scheduling algorithms?
10. Explain briefly the performance of demand paging with necessary examples.
11. Compare and contrast chained allocation with indexed allocation technique of file allocation.
12. A process refers to 5 pages, A, B, C, D, and E in the order- A; B; C; D; A; B; E; A;
    B; C; D; E. If the page replacement algorithm is LRU, calculate the number of page
    faults with empty frames of size 4?



# MODEL SET 5

## Section 'A'

**Long Answer Questions**

**Attempt any two questions.** **(2 × 10=20)**

1. Define process. How many different states a process has? Explain when a process changes the state with a state diagram?
2. Explain banker's algorithm for deadlock avoidance with an example.
3. Consider that a disk drive has 5,000 cylinders, numbered 0 to 4,999. The drive is currently serving request at cylinder 143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order, is:
   86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130
   Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all pending requests for each of the following disk scheduling algorithms?
   a. FCFS b. SSTF
   c. SCAN d. C-SCAN
   e. LOOK f. C-LOOK

**Short Questions**

**Attempt any eight questions.** **(8 × 5 = 40)**

4. List the various disk space allocation strategies. Explain clearly the contiguous allocation technique.
5. What is deadlock? Explain the conditions that lead to deadlock.
6. Suppose that a disk drive has 5000 cylinders numbered 0 to 4999. The drive is currentlyserving a request at cylinder 143. The queue of pending requests in FIFO order86,1470,913,1774,948,1509, 1022, 1750, 130 starting from current head position. What is the total distance that disk arm moves to satisfy the entire pending request for FCFS and SSTF disk scheduling algorithm?
7. Describe Resource-Allocation graph? Explain how resource graph can be used for detecting deadlocks.
8. State the purpose of short-term, medium-term and long term schedulers. Also discuss the differences among them.
9. What is Semaphore? How can we achieve the synchronization using semaphore for producer consumer problem?
10. Define Essential properties of Distributed Operating Systems.
11. What system calls have to be executed by a command interpreter or shell in order to start a new process? Explain briefly.
12. A system has 3 devices D1, D2 and D3 and 3 processes P1, P2, and P3. P1 is holdingD1 and waiting for D3. P2 is holding D2 and waiting for D1. P3 is holding D3 and waiting for D2.Draw resource allocation graph and wait-for graph. Is the system in deadlock state or not? Explain.

Scanned by TapScanner

# MODEL SET 6

## Section 'A'

**Long Answer Questions**

**Attempt any two questions.** **(2 × 10=20)**

1. Explain how file allocation table (FAT) manages the files. Maintain the merits and demerits of FAT system. A 200 GB disk has 1-KB block size, calculate the size of the file allocation table if each entry of the table to be 3 bytes.
2. Define the term semaphore. How does semaphore help in dining philosophers problem?
3. Suppose that a disk has 100 cylinders, numbered 0 to 99. The drive is currently serving a request at cylinder 43 and previous request was at cylinder 25. The queue of pending queue, in FIFO order is:

   86, 70, 13, 74, 48, 9, 22, 50, 30, 100, 33

   Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all pending request for each of the following disk scheduling algorithms?

   a. FCFS
   b. SCAN

**Short Questions**

**Attempt any eight questions.** **(8 × 5 = 40)**

4. What is operating system? Differentiate between time sharing and real time operating system.
5. Why thread is necessary? In which circumstances user-level thread is better than kernel level thread?
6. Explain about hierarchical directory systems with diagrammatic examples.
7. What do you mean by interrupt? Explain the working mechanism of interrupt controller.
8. Define the term indefinite postponement. How does it differ from deadlock?
9. Explain the mapping of virtual address to real address under segmentation. Compare the throughput (overall performance) of SCAN with SSTF.
10. Explain the disk management with example.
11. Does window have any concept of process hierarchy? How does parent control the child?
12. How many bits would be needed to store the free-space list under the following condition if a bitmap were used to implement?
    a. 500000 blocks total and 200000 free blocks
    b. 1000000 blocks total and 0 free blocks



# MODEL SET 7

## Section 'A'

**Long Answer Questions**

**Attempt any two questions.** **(2 × 10=20)**

1. Consider the following page reference strings:

   1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6.

   How many page faults would occur for the following replacement algorithm, assumingthree,four frames? Remember that all frames are initially empty, so your first uniquepages will cost one fault each.

   a. LRU replacement b. Optimal replacement
2. Explain the reader's writer's problem and its solution using the concept of semaphores.
3. What is deadlock? Explain various conditions for deadlock. Discuss the Bankers algorithm of multiple resources for avoidance of deadlock with suitable example.

**Short Questions**

**Attempt any eight questions.** **(8 × 5 = 40)**

4. How physical address is generated from logical address? Explain the process of system call with suitable diagram.
5. How does process differ from program? Explain process state with the help of block diagram.
6. Explain how multithreading improve performance over a single threaded solution.
7. Describe how Peterson's solution preserves mutual exclusion in process CR execution.
8. What is Fragmentation? Differentiate between internal and external fragmentation.
9. How does DMA increase system concurrency? How it complicate hardware design?
10. Given a references to the following pages by program,

    1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 3, 6, 9

    How many page faults will occur if the program has three page frames available to it and users LRU replacement?
11. What are the differences between the trap and interrupt? What is the use of each function?
12. Suppose that the disk drive has 50 cylinders, numbered from 0 to 49. The drive currently saving the request at cylinder 20 and the previous request was at cylinder 25. The queue of pending request s 10, 22, 20, 2, 40, 6 and 38 in the order. A seek takes 6msec per cylinder moved. How much seek time is needed for the following disk scheduling algorithms?

    a. First-come, First-served b. Shortest Seek time first
    c. SCAN d. LOOK

Scanned by TapScanner

# MODEL SET 8

## Section 'A'

**Long Answer Questions**

**Attempt any two questions.** **(2 × 10=20)**

1. Consider the following set of processes with the length of the CPU burst time given in milliseconds

| Process | Burst Time | Priority |
|---|---|---|
| P1 | 10 | 3 |
| P2 | 1 | 1 |
| P3 | 2 | 3 |
| P4 | 1 | 4 |
| P5 | 5 | 2 |

The processes are assumed to have arrived in the order p1, p2, p3, p4, p5 all at time 0.

a. Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, anon pre-emptive priority (a smaller priority number implies a higher priority) and RR (quantum=1)scheduling.

b. What is the turnaround time of each process for each of the scheduling algorithms in part?

c. What is the waiting time of each process for each of the scheduling algorithms in part?

d. Which of the schedules in part a results in the minimal average waiting time?

2. Define Monitor? Compare it with semaphore. Explain in detail a monitor with notify and broadcast using an example.

3. What is system call? Explain in detail the types of system calls provided by a typical operating system?

**Short Questions**

**Attempt any eight questions.** **(8 × 5 = 40)**

4. Define file and directories. Explain about protection mechanism.

5. State whether you agree or disagree with the following claim: The Banker's algorithm may fail to avoid deadlock if a resource breaks down. Justify your answer.

6. What are the main characteristics of Real Time Operating System?

7. Explain how protection can be ensured using paging?



8. Explain why the "principle of locality" is crucial to the use of virtual memory? What is accomplished by page buffering?

9. Consider the following page reference string

   1,2,3,4,5,3,4,1,6,7,8,7,8,9,7,8,9,5,4,5,4,2

   With four frames, how many page faults would occur for the FIFO, Optimal page replacement algorithms? Which algorithm is efficient? (Assume all frame are initially empty).

10. A system has four processes and five allocatable resources. The current allocation and maximum needs are as follows:

| | Allocated | Maximum | Available |
|---|---|---|---|
| Process A | 1 0 2 1 1 | 1 1 2 1 3 | 0 0 x 1 1 |
| Process B | 2 0 1 1 0 | 2 2 2 1 0 | |
| Process C | 1 1 0 1 0 | 2 1 3 1 0 | |
| Process D | 1 1 1 1 0 | 1 1 2 2 1 | |

What is the smallest value of x for which this is a safe state?

11. What does Contiguous Allocation requires? Write down their advantages and disadvantages.

12. Suppose that a disk drive has 5000 cylinders numbered 0 to 4999. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order is:

    85, 1465, 920, 1784, 948, 1510, 1025, 1745, 128

    Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk scheduling algorithms?

    a. SSTF  b. SCAN
    c. C-SCAN  d. C-LOOK
    e. LOOK  f. FIFO

❑❑❑

Scanned by TapScanner

# Artificial Intelligence

*Credits:* 3
*Lecture Hours:* 48

**Course Objectives**

This module aims to provide the students with the basic foundation on concepts of searching and knowledge representation in AI systems. The key objective is to make students more pragmatic in knowledge of AI by giving its applications like designing and training Artificial Neural Networks along with additional laboratory works.

**Course Description**

Introduction, Agents and Environments, Informed and Uninformed Search, Knowledge Representation, Learning, Applications of AI, Production Systems, Uncertainty in AI.

**Course Details**

**Unit 1: Introduction** — LH 4

1.1 What is AI?
- 1.1.1 Turing test approach: Chinese room argument
- 1.1.2 Cognitive approach
- 1.1.3 Laws of thought approach
- 1.1.4 Rational agent approach

1.2 Difference between AI and Omniscience

**Unit 2: Agents and Environments** — LH 7

2.1 Agent, Rational agent, and Intelligent Agent
2.2 Relationship between agents and environments
2.3 Environments and its properties
2.4 Agent structures
- 2.4.1 Simple reflex agents
- 2.4.2 Model-based reflex agents
- 2.4.3 Goal-based agents
- 2.4.4 Utility-based agents
- 2.4.5 Learning agents

2.5 Performance evaluation of agents: PEAS description

**Unit 3: Informed and Uninformed Search** — LH 8

3.1 Why search in AI?
3.2 Blind search (Un-informed search)
- 3.2.1 Breadth first search (BFS)
  - **Variations:** Uniform cost search
- 3.2.2 Depth first search (DFS)
  - **Variations:** Depth limited search, Iterative deepening DFS

3.3 Heuristic search (Informed search)
- 3.3.1 Hill climbing
  - 3.3.1 The Foothills Problem
  - 3.3.2 The Plateau Problem
  - 3.3.3 The Ridge Problem
- 3.3.2 Greedy (Best-first) search
- 3.3.3 A* algorithm (search)
- 3.3.4 Means-Ends Analysis: Household ROBOT, Monkey Banana Problem



3.4 General Problem Solving (GPS): Problem solving agents
- 3.4.1 Constraint satisfaction problem
- 3.4.1 Constraint Satisfaction Search
- 3.4.2 AND/OR trees
- 3.4.3 The bidirectional search
- 3.4.4 Cryptoarithmatic

3.5 Game playing and AI
- 3.2.1 Game Trees and Minimax Evaluation
- 3.2.2 Heuristic Evaluation
- 3.2.3 Min-max algorithm (search)
- 3.2.4 Min-max with alpha-beta
- 3.2.5 Games of chance
- 3.2.6 Game theory

**Unit 4: Knowledge Representation** — LH 8

4.1 Logic
- 4.1.1 Propositional Logic
  - 4.1.1.1. Syntax, semantics, and properties
  - 4.1.1.2. Conjunctive Normal Form (CNF)
  - 4.1.1.3. Disjunctive Normal Form (DNF)
  - 4.1.1.4. Inference Rules
  - 4.1.1.5. Resolution
- 4.1.2 Prehdicate Logic
  - 4.1.1.1. First-Order Predicate Logic (FOPL)
  - 4.1.1.2. Syntax and semantics in FOPL
  - 4.1.1.3. Quantifiers
  - 4.1.1.4. Clausal Normal Form
  - 4.1.1.5. Resolution
- 4.1.3 Fuzzy Logics

4.2 Semantic networks (nets): Introduction, and examples

**Unit 5: Learning** — LH 6

5.1 Why learning?
5.2 Supervised (Error based) learning
- 5.2.1 Gradient descent learning: Least Mean Square, Back Propagation algorithm
- 5.2.2 Stochastic learning

5.3 Unsupervised learning
- 5.3.1 Hebbian learning algorithm
- 5.3.2 Competitive learning

5.4 Reinforced learning (output based)
5.5 Genetic algorithms: operators

**Unit 6: Applications of AI** — LH 5

6.1 Artificial Neural Networks (ANN)
- 6.1.1 Neural Networks (NN) and ANN
- 6.1.2 Activation functions: unit (unary and binary), ramp, piecewise linear, & sigmoid
- 6.1.3 Training and testing: Basic concept
- 6.1.4 Mc-Colloch-Pits neuron model
  - 6.1.5.1. Realization of AND, OR, NOT, and XOR gates

Scanned by TapScanner

6.1.5 Neural network architectures
6.1.6.1. Single layer feed-forward architecture: ADALINE, Perceptron NN
6.1.6 Applications of ANN
6.2 Natural Language Processing (NLP)
6.2.1 Fundamentals of language processing

**Unit 7: Production systems** LH 4
7.1 Strong Methods vs Weak Methods
7.2 Advantages of Production Systems
7.3 Production Systems and inference methods
7.3.1 Conflict resolution strategies
7.3.2 Forward chaining
7.3.3 Backward chaining

**Unit 8: Uncertainty in AI** LH 3
8.1 Fuzzy sets
8.2 Fuzzy logic
8.3 Fuzzy inferences
8.4 Probability theory and uncertainty

**Unit 9: Expert Systems Human and Machine experts** LH 3
9.1 Characteristics of expert systems
9.2 Knowledge engineering
9.3 Knowledge acquisition
9.4 Classic expert system
9.4.1 DENDRAL
9.4.2 MYCIN
9.4.3 EMYCIN
9.5. Case based reasoning

**Lab Task:**
Students are required to carry out at least 6 lab tasks on predicate calculus, searching and neural networks using **ProLog** and C/C++/Java. Some of the lab tasks may be on:
1 Relationship programs (e.g. mother, father, brother etc)
2 Recursive programs: Factorial, Fibonacci series etc
3 Ancestor programs
4 Tower of Hanoi (TOH) program
5 Monkey banana problem
6 Realization of logic gates (using C/C++/Java)

**References**
Russel S. and Norvig P., ***Artificial Intelligence: A modern Approach***, Prentice hall
Ritch and Knight, ***Artificial Intelligence***, Prentice hall
Dan W. Patterson, ***Artificial Intelligence***
Artificial Intelligence in the 21st century, . .phen Lucci, Danny Kopec, Mercury Learning and Information
Artificial Intelligence: **Foundations of computational Agents**, David L. Poole, Alan K. Mackworth, first edition, Cambridge University Press



# TRIBHUVAN UNIVERSITY

**Institution of Science and Technology**

Course Title: Artificial Intelligence — Full Marks: 60
Course No: CSC261 — Pass Marks: 24
Level: B. Sc CSIT Second Year/ Fourth Semester — Time: 3 Hrs

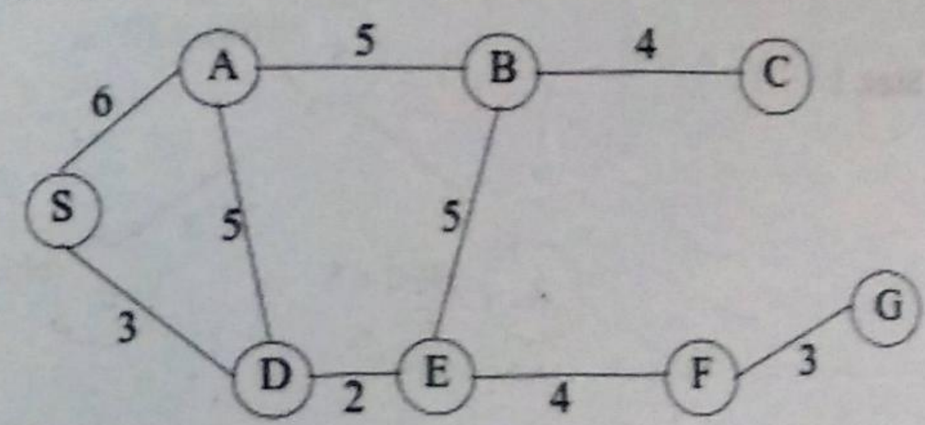
## T.U. MODEL QUESTIONS-ANSWERS SET I

### Section A

**Long Answer Questions**

**Attempt any two questions.** [2*10=20]

1. **What do you mean by heuristic search? Given following state space representation, show how greedy best first and A* search is used to find the goal state.** [2+8]




**S is the start state and G is the goal state. The heuristics of the states are h(S) = 12,**
**h(A) = 8, h(D) = 9, h(B) = 7, h(E) = 4, h(C) = 5 , h(F) = 2, h(G)= 0.**

**Answer:** Heuristic Search uses domain-dependent (heuristic) information in order to search the space more efficiently. It is also known as informed search. In other words, Heuristic search refers to a search strategy that attempts to optimize a problem by iteratively improving the solution based on a given heuristic function or a cost measure. In this search method, it might not always find the best solution but is guaranteed to find a good solution in reasonable time. By sacrificing completeness it increases efficiency. It is useful in solving tough problems which

- Could not be solved any other way.

Scanned by TapScanner

- Solutions take an infinite time or very long time to compute.

**Greedy best first**

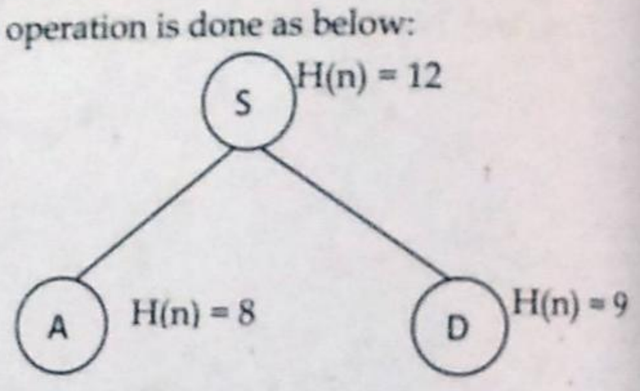
Straight Line distances to node G (goal node) from other nodes is given below:

S→ G = 12
A→ G = 8
D→ G = 9
B→ G = 7
E→ G = 4
C→G = 5
F→ G = 2
G→G = 0

Let H(n)= Straight Line distance

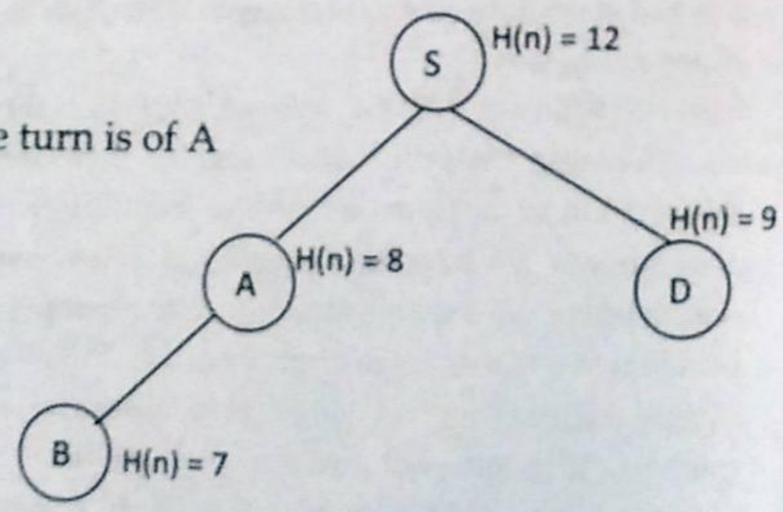
Now Greedy Search operation is done as below:

**Step 1**




**Step 2:**

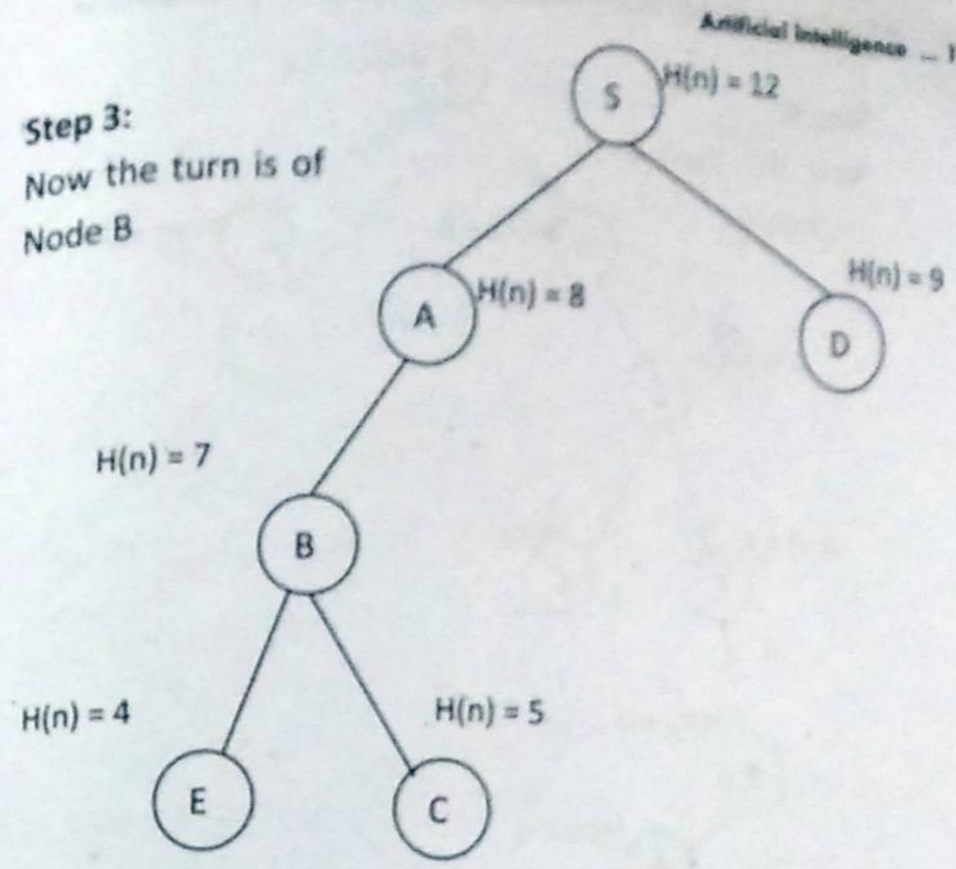
Now the turn is of A





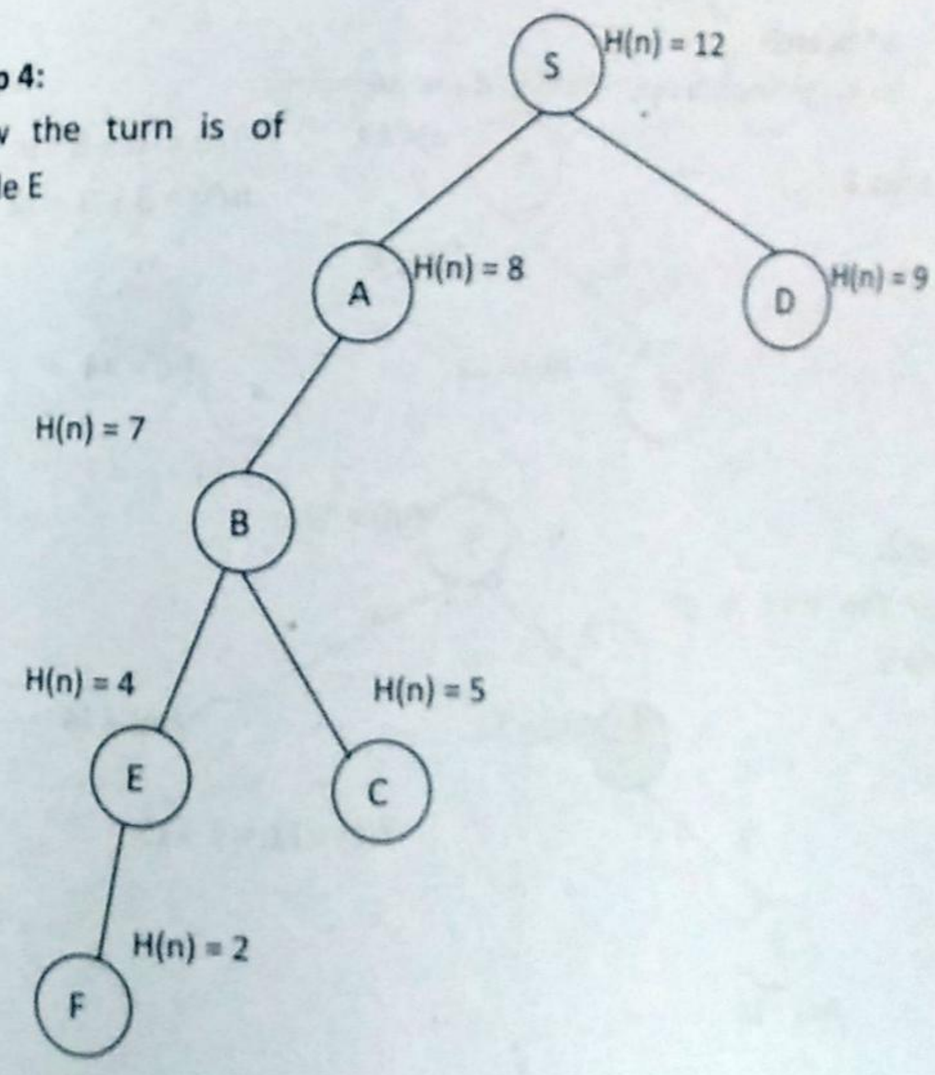

**Step 3:**

Now the turn is of Node B



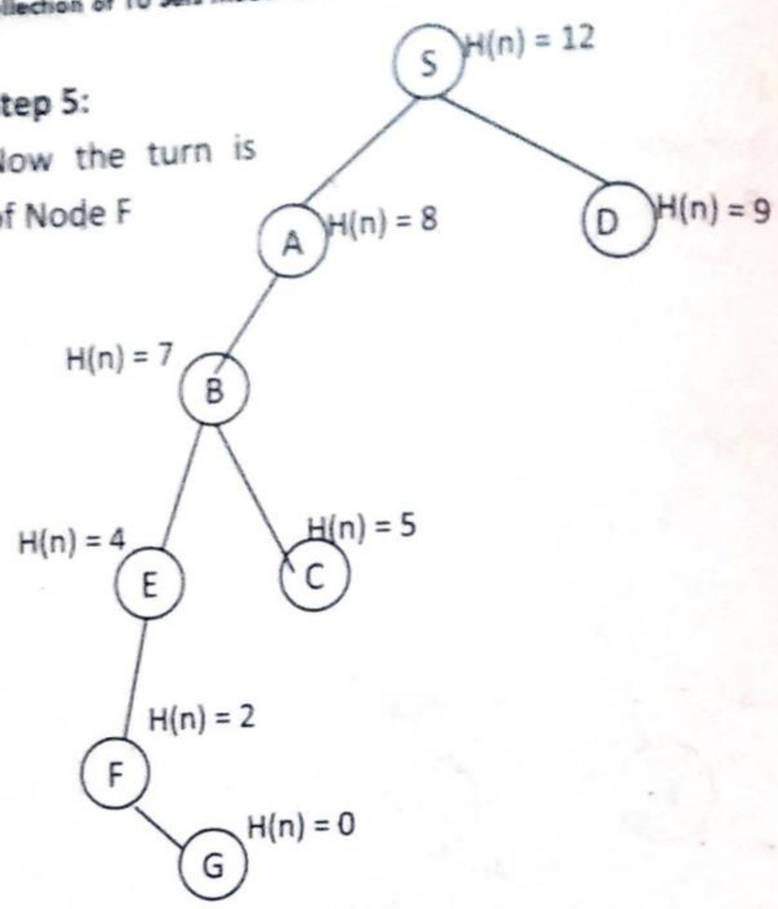

**Step 4:**

Now the turn is of Node E




Scanned by TapScanner

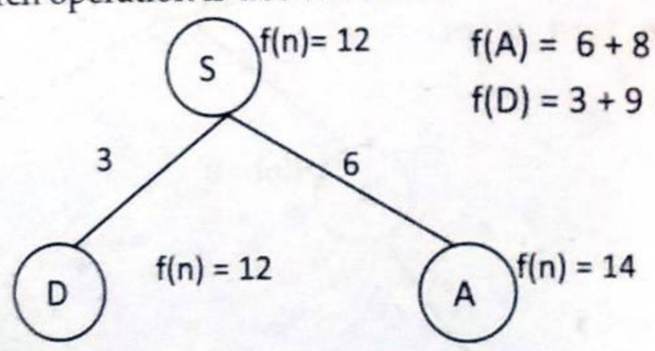
**Step 5:**
Now the turn is of Node F




**A* Search**

Now A* Search operation is done as below

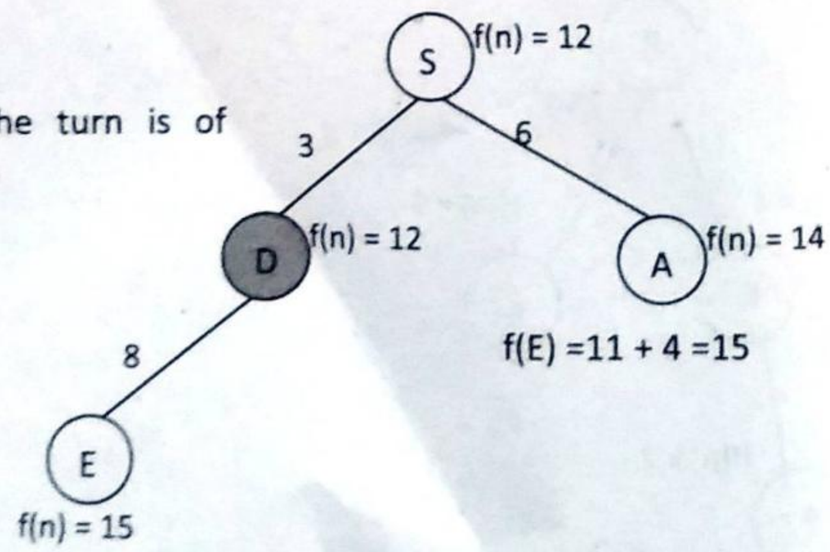
**Step 1**




f(A) = 6 + 8 =14

f(D) = 3 + 9 =12

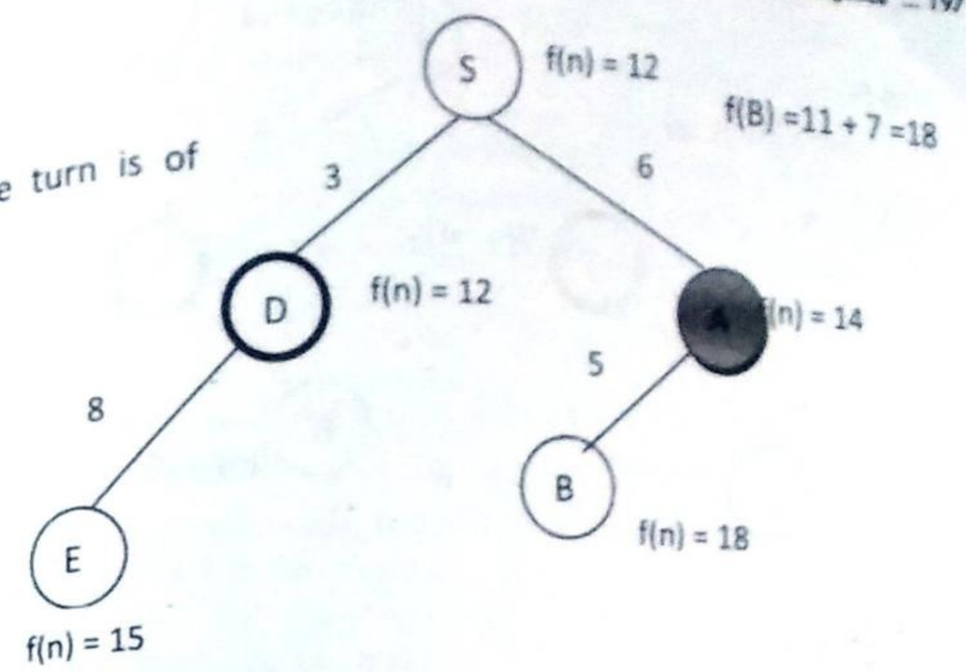
**Step 2:**
Now the turn is of node D




f(E) =11 + 4 =15

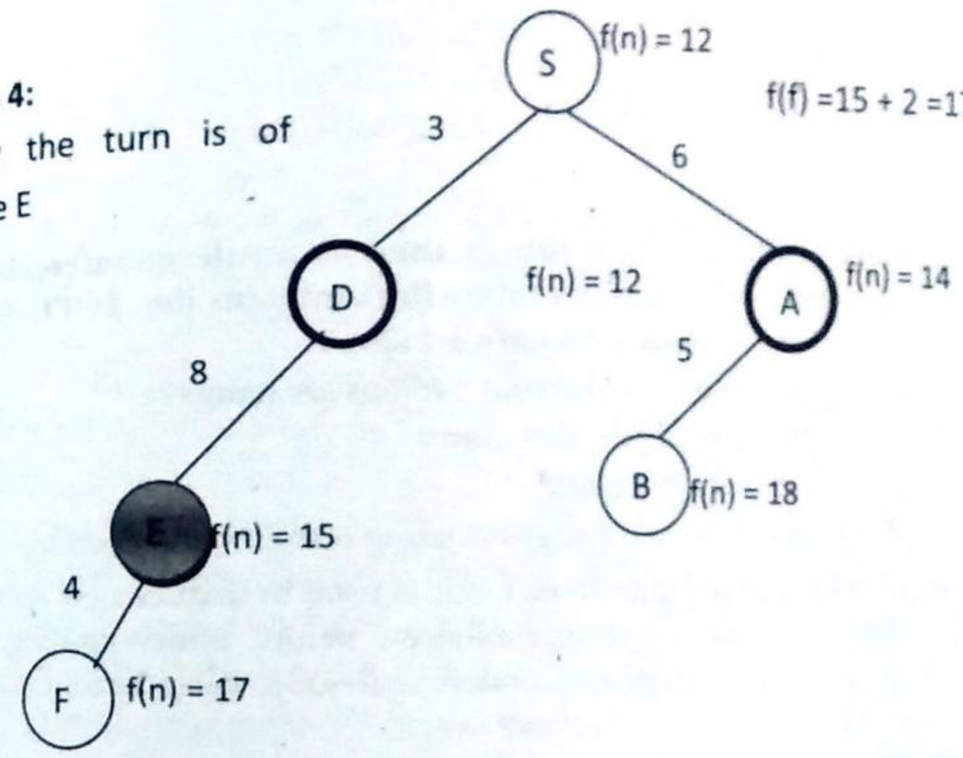


**Step 3:**
Now the turn is of node A




f(B) =11 + 7 =18

**Step 4:**
Now the turn is of node E

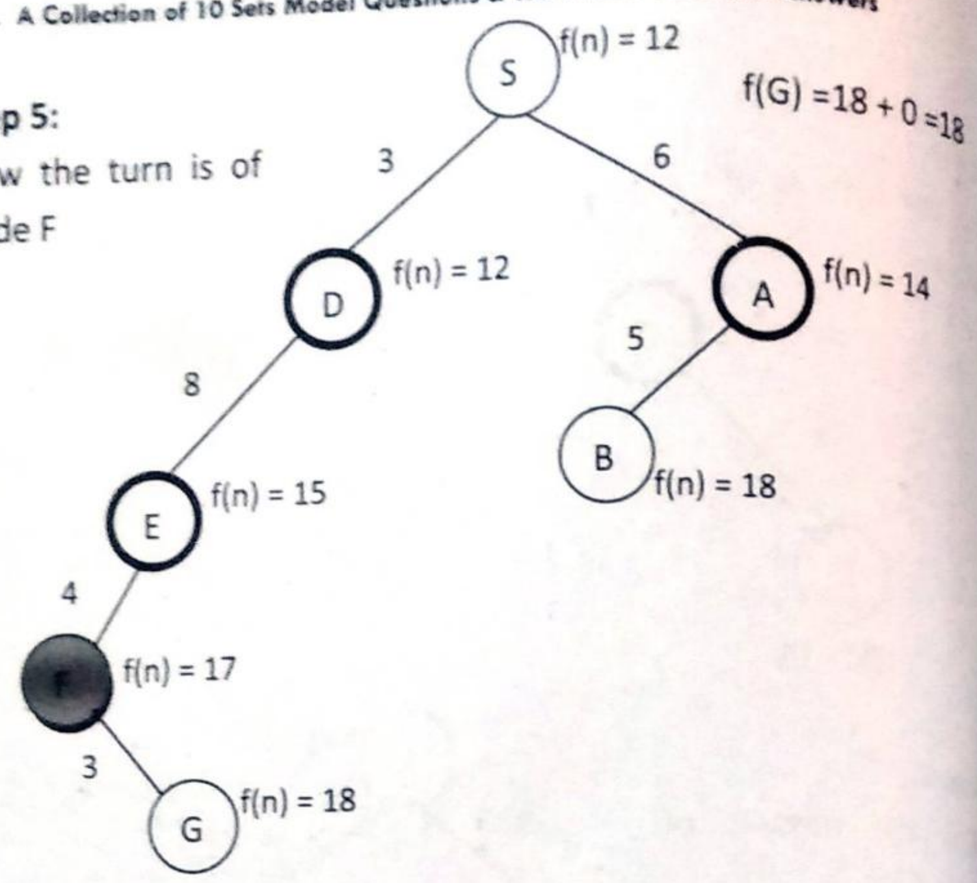



f(f) =15 + 2 =17

Scanned by TapScanner

**Step 5:**

Now the turn is of node F




2. **How resolution algorithm is used as a rule of inference in predicate logic? Convert following sentences into FOPL.** [4+6]
   - All over smart person's are stupid
   - Children's of all stupid persons are naughty
   - Roney is Children of Harry
   - Harry is over smart

   Prove that "Roney is naughty" using resolution algorithm.

**Answer:** Inference in First-Order Logic is used to deduce new facts or sentences from existing sentences. Before understanding the FOL inference rule, let's understand some basic terminologies used in FOL.

**Resolution**

Resolution is a theorem proving technique that proceeds by building refutation proofs, i.e., proofs by contradictions. It was invented by a Mathematician John Alan Robinson in the year 1965. Resolution is used, if there are various statements are given, and we need to prove a conclusion of those statements. Unification is a key concept in proofs by resolutions. Resolution is a single inference rule which can efficiently operate on the conjunctive normal form or clausal form.

**Clause:** Disjunction of literals (an atomic sentence) is called a clause. It is also known as a unit clause.



**Conjunctive Normal Form:** A sentence represented as a conjunction of clauses is said to be conjunctive normal form or CNF.

The resolution rule for first-order logic is simply a lifted version of the propositional rule. Resolution can resolve two clauses if they contain complementary literals, which are assumed to be standardized apart so that they share no variables.

$$l_1 \vee \ldots\ldots \vee l_k, \qquad m_1 \vee \ldots\ldots \vee m_n$$

$$\text{SUBST}(\theta, l_1 \vee \ldots\ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots\ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n)$$

Where li and mj are complementary literals.

This rule is also called the binary resolution rule because it only resolves exactly two literals.

**Steps**

1. Negate the statement to be proved
2. Convert given fact into FOPL
3. Convert FOPL into CNF
4. Draw resolution graph

Numerical parts,

- Let's assume all are instances of person,

$\forall x$ (oversmart(x)) $\wedge$ person(x))$\rightarrow$stupid(x)

$\forall x$ (oversmart(x))$\rightarrow$ stupid(x)

$\forall y$ (Children(y, x))$\wedge$ stupid(x)$\rightarrow$naughty(y)

Children (Roney, Harry)

Oversmart (Harry)

Naughty (Roney)

Now express relations in the form $\neg \alpha \vee \beta$

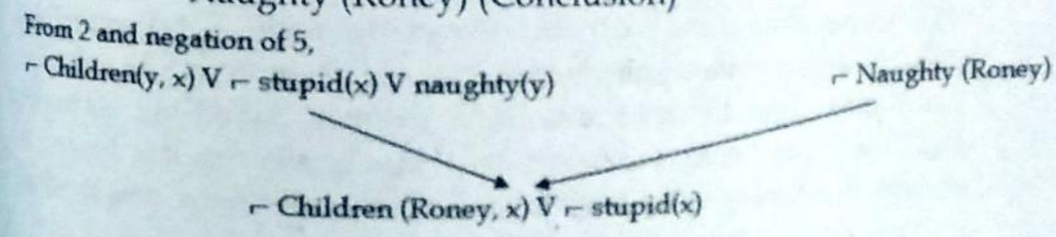
$\forall x$ ($\neg$ oversmart(x)) V stupid(x)

$\forall y$ ($\neg$ Children(y, x)) $\wedge$ stupid(x) V naughty(y)

Now we have following statements,

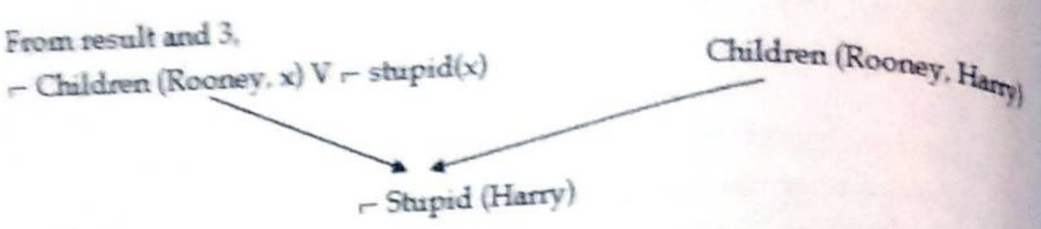
1. $\neg$ oversmart(x) V stupid(x)
2. $\neg$ Children(y, x)) V $\neg$ stupid(x) V naughty(y)
3. Children (Roney, Harry)
4. Oversmart (Harry)
5. Naughty (Roney) (Conclusion)

From 2 and negation of 5,

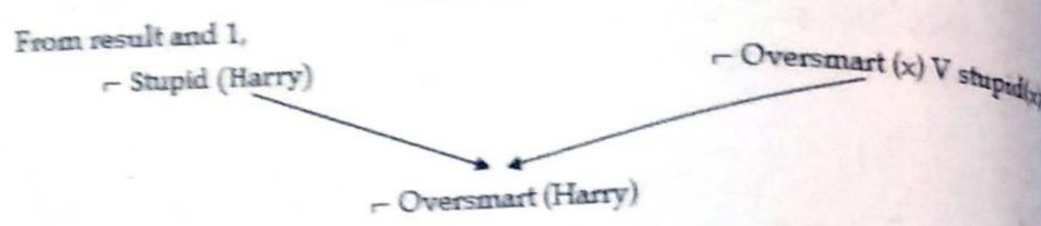



Scanned by TapScanner

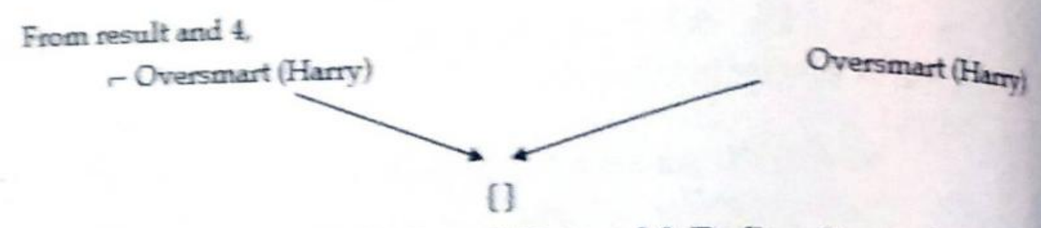
From result and 3,




From result and 1,




From result and 4,




3. **What is Artificial Neural Network? Define its mathematical model. Discuss how back propagation algorithm is used to train ANN?** **[1+2+6]**

**Answer:** An artificial neuron network (ANN) is a computational model based on the structure and functions of biological neural networks. Information that flows through the network affects the structure of the ANN because a neural network changes - or learns, in a sense - based on that input and output. ANNs are considered nonlinear statistical data modeling tools where the complex relationships between inputs and outputs are modeled or patterns are found. ANN is also known as a neural network.

**Backpropagation**

It is a supervised learning method, and is an implementation of the Delta rule. It requires a teacher that knows, or can calculate, the desired output for any given input. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop). The term is an abbreviation for "backwards propagation of errors". Backpropagation requires that the activation function used by the artificial neurons (or "nodes") is differentiable.

Backpropagation networks are necessarily multilayer perceptrons (usually with one input, one hidden, and one output layer). In order for the hidden layer to serve any useful function, multilayer networks must have non-linear activation functions for the multiple layers: a multilayer network using only linear activation functions is equivalent to some single layer, linear network.



**Algorithm**

**Input:**

*D*, a data set consisting of the training tuples and their associated target value;

*l*, the learning rate;

*network*, a multilayer feed forward network

**Output**: A trained neural network

**Method:**

Initialize all weights and biases in network;

**while** terminating condition is not satisfied

{

**for** each training sample X in *D*

{

// Propagate the inputs forward:

**for** each input layer j

$O_j = I_j$; //output of an input unit is its actual input value

**for** each hidden or output layer unit j

{

$I_j = \sum w_{ij}O_i + \theta j$; //compute the net input of unit j with respect to previous input layer, i

$O_j = 1$

$1+e^{-Ij}$;

}

// compute the output of each unit j

// Backpropagate the errors:

**for** each unit j in the output layer

$Err_j = O_j(1 - O_j)(T_j-O_j)$; // compute the error

**for** each unit j in the hidden layers

$Err_j = O_j(1 - O_j)\sum_k Err_k w_{jk}$; // compute the error

**for** each weight $w_{ij}$ in network

{

$\Delta w_{ij} = (l)Err_jO_i$; // weight increment

$w_{ij} = w_{ij}+\Delta w_{ij}$; // weight update

}

**for** each bias _j in network

{

$\Delta\theta j = (l)Err_j$ ; // bias increment

$\theta_j = \theta j+\Delta\theta_j$; // bias update

}}}

Scanned by TapScanner

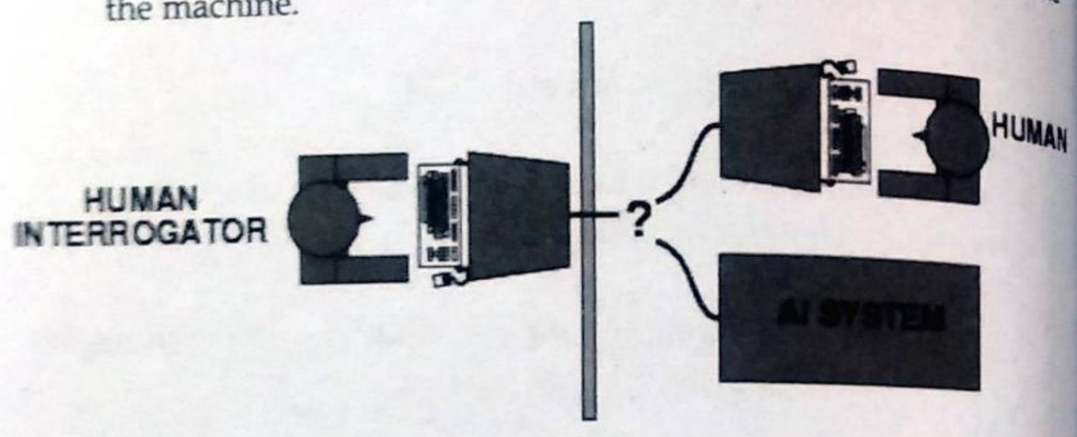
## Section B

**Short Answer Questions**

**Attempt any eight questions.** [8*5=40]

4. Describe how Turing Test is used to define AI as acting humanly?

**Answer:** The Turing test, proposed by Alan Turing (1950) was designed to convince the people that whether a particular machine can think or not. He suggested a test based on in distinguish ability from undeniably intelligent entities- human beings. The test involves an interrogator who interacts with one human and one machine. Within a given time the interrogator has to find out which of the two the human is, and which one the machine.




The computer passes the test if a human interrogator after posing some written questions, cannot tell whether the written response come from human or not. To pass a Turing test, a computer must have following capabilities:

- **Natural Language Processing:** Must be able to communicate successfully in English
- **Knowledge representation:** To store what it knows and hears.
- Automated reasoning: Answer the Questions based on the stored information.
- **Machine learning**: Must be able to adapt in new circumstances.

Turing test avoid the physical interaction with human interrogator. Physical simulation of human beings is not necessary for testing the intelligence.

**The total Turing test** includes video signals and manipulation capability so that the interrogator can test the subject's perceptual abilities and object manipulation ability. To pass the total Turing test computer must have following additional capabilities:



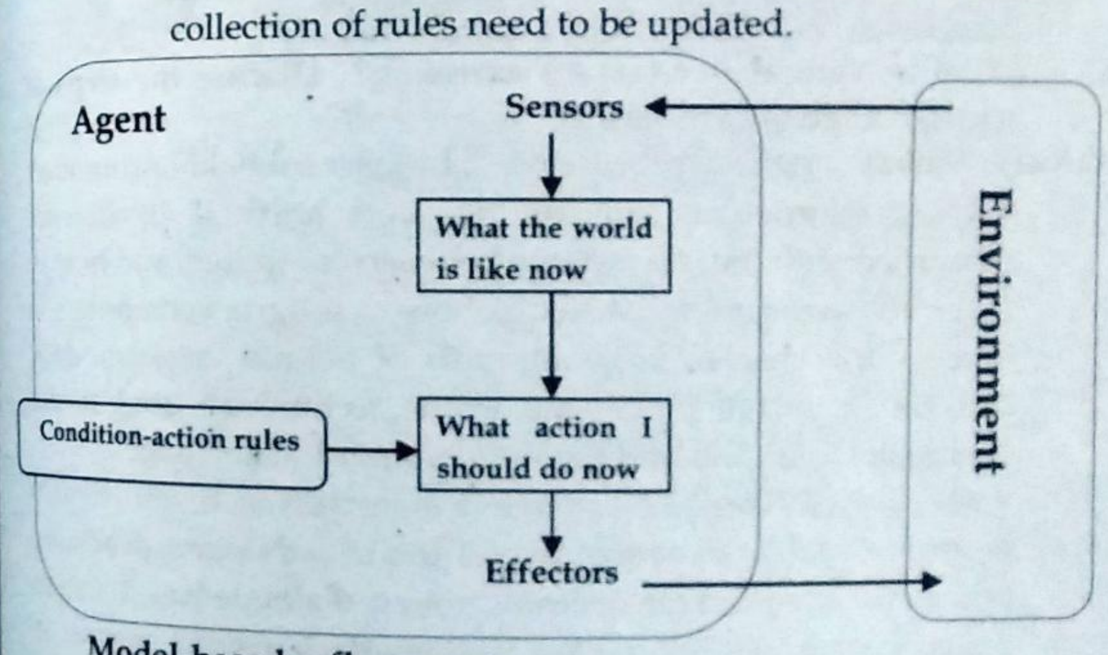
- Computer Vision: To perceive objects
- Robotics: To manipulate objects and move

5. **Differentiate between model based and simple reflex agent with an example.**

**Answer: Simple reflex agents**

Simple reflex agents ignore the rest of the percept history and act only on the basis of the current percept. Percept history is the history of all that an agent has perceived till date. The agent function is based on the condition-action rule. A condition-action rule is a rule that maps a state i.e., condition to an action. If the condition is true, then the action is taken, else not. This agent function only succeeds when the environment is fully observable. For simple reflex agents operating in partially observable environments, infinite loops are often unavoidable. It may be possible to escape from infinite loops if the agent can randomize its actions. Problems with Simple reflex agents are:

- Very limited intelligence.
- No knowledge of non-perceptual parts of state.
- Usually too big to generate and store.
- If there occurs any change in the environment, then the collection of rules need to be updated.



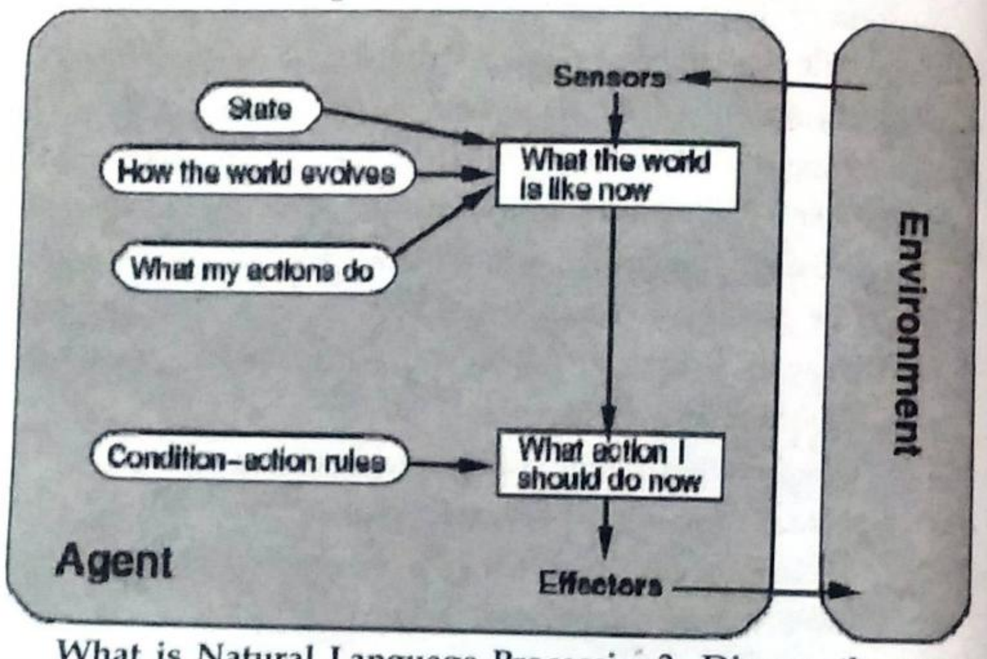

**Model-based reflex agents**

It works by finding a rule whose condition matches the current situation. A model-based agent can handle partially observable environments by use of model about the world. The agent has

Scanned by TapScanner

to keep track of internal state which is adjusted by each percept and that depends on the percept history. The current state is stored inside the agent which maintains some kind of structure describing the part of the world which cannot be seen. Updating the state requires the information about:

- How the world evolves in-dependently from the agent, and
- How the agent actions affects the world.




6. **What is Natural Language Processing? Discuss the steps of natural language processing.** [1+4]

**Answer:** Natural language processing (NLP) is a subfield of computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data. Natural Language Processing is the technology used to aid computers to understand the human's natural language.

Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions, when you want to hear decision from a dialogue based clinical expert system, etc. The field of NLP involves making computers to perform useful tasks with the natural languages humans use. The input and output of an NLP system can be –

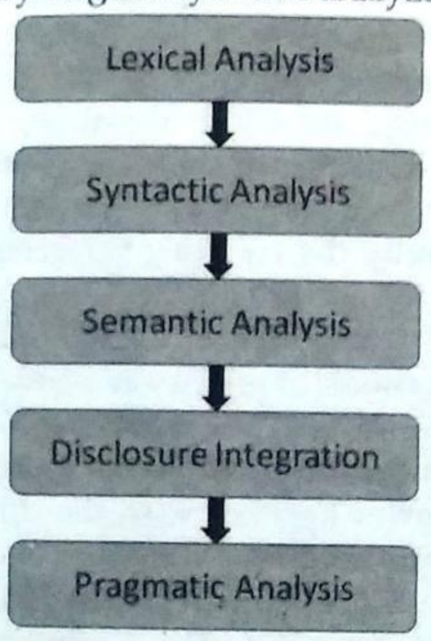
- Speech
- Written Text



**Steps in NLP**

There are general five steps –

1. Lexical Analysis – It involves identifying and analyzing the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of txt into paragraphs, sentences, and words.
2. **Syntactic Analysis (Parsing)** – It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as "The school goes to boy" is rejected by English syntactic analyzer.




3. **Semantic Analysis** – It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyzer disregards sentence such as "hot ice-cream".
4. **Discourse Integration** – the meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence.
5. **Pragmatic Analysis** – During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

Scanned by TapScanner

7. How belief networks are constructed? Consider the probability of having cloudy is 50%. The probability that it will rain given the conditions it will be cloudy and if it is winter is 30%. The probability of being winter is 50%. The probability that it will be shiny is 70%. Now construct a belief network for this example. [2+3]

**Answer:** Bayesian networks are widely used in medical and engineering diagnostics under incomplete and imperfect information, in data classification systems, automatic speech recognition, marketing, business, and many other activities. In the general case, BNs make it possible to reproduce cause-effect relationships among events and to determine event probability if new information on changes in the state of any node (variable) of the network becomes available. The degree of expediency of applying this method of modeling and probabilistic inference depends on the ability to formulate the problem correctly, to select variables of the process that characterize its dynamics or statics in a sufficient degree, to find necessary data and use them for network learning, and formulate correctly the resulting inference using the network constructed.

A Bayesian network consists of the following:

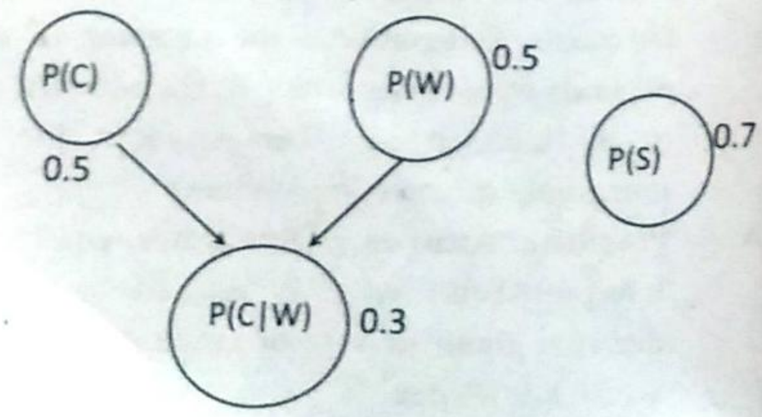
- A set of variables and a set of directed edges between variables.
- Each variable has a finite set of mutually exclusive states.
- The variables together with the directed edges form an acyclic directed graph (DAG).
- To each variable A with parents B1. . . Bn, a conditional probability table P(A | B1, . . . , Bn) is attached.

Let, probability of cloudy=P(C) = 50%=0.5

Probability that it will rain given the conditions it will be cloudy and if it is winter is = P(C | W) = 30%=0.3

The probability of being winter = P(W) = 50% = 0.5

Probability that it will be shiny = 70% = 0.7






8. What is expert system? Explain the major components of Expert System? [1+4]

**Answer:** The expert systems are the computer applications developed to solve complex problems in a particular domain, at the level of extra-ordinary human intelligence and expertise.

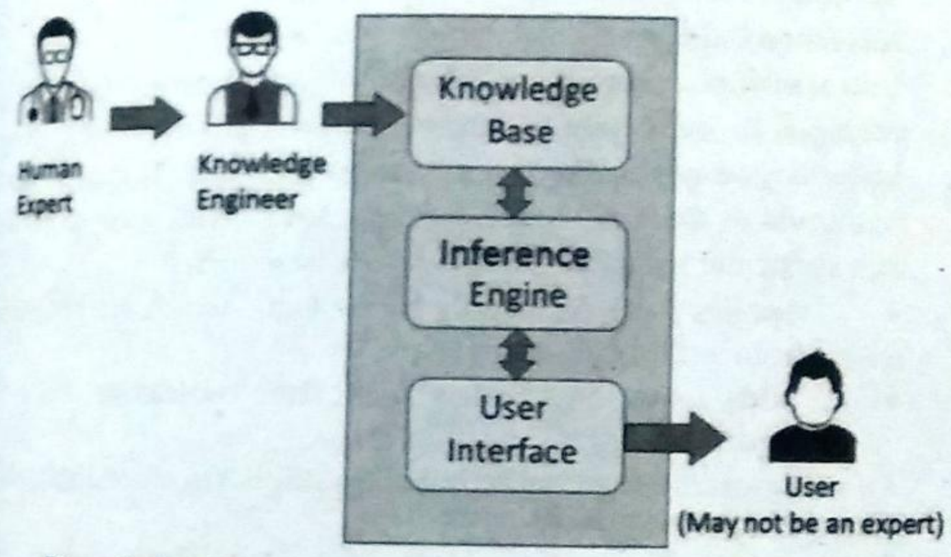
**Characteristics of Expert Systems**

- High performance
- Understandable
- Reliable
- Highly responsive

**Components of Expert Systems**

The components of ES include –

a) Knowledge Base
b) Inference Engine
c) User Interface




**Knowledge Base**

It contains domain-specific and high-quality knowledge. Knowledge is required to exhibit intelligence. The success of any ES majorly depends upon the collection of highly accurate and precise knowledge.

The data is collection of facts. The information is organized as data and facts about the task domain. Data, information, and past experience combined together are termed as knowledge.

**Components of Knowledge Base**

The knowledge base of an ES is a store of both, factual and heuristic knowledge.

Scanned by TapScanner

a) Factual Knowledge – It is the information widely accepted by the Knowledge Engineers and scholars in the task domain.

b) Heuristic Knowledge – It is about practice, accurate judgement, one's ability of evaluation, and guessing.

The success of any expert system majorly depends on the quality, completeness, and accuracy of the information stored in the knowledge base. The knowledge base is formed by readings from various experts, scholars, and the Knowledge Engineers. The knowledge engineer is a person with the qualities of empathy, quick learning, and case analyzing skills. He acquires information from subject expert by recording, interviewing, and observing him at work, etc. He then categorizes and organizes the information in a meaningful way, in the form of IF-THEN-ELSE rules, to be used by interference machine. The knowledge engineer also monitors the development of the ES.

**Inference Engine**

Use of efficient procedures and rules by the Inference Engine is essential in deducting a correct, flawless solution. In case of knowledge-based ES, the Inference Engine acquires and manipulates the knowledge from the knowledge base to arrive at a particular solution. In case of rule based ES, it –

- Applies rules repeatedly to the facts, which are obtained from earlier rule application.
- Adds new knowledge into the knowledge base if required.
- Resolves rules conflict when multiple rules are applicable to a particular case.

To recommend a solution, the Inference Engine uses the following strategies –

- Forward Chaining
- Backward Chaining

**User Interface**

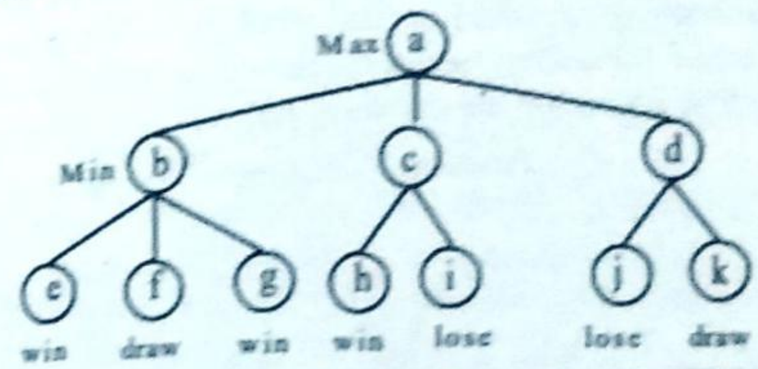
User interface provides interaction between user of the ES and the ES itself. It is generally Natural Language Processing so as to be used by the user who is well-versed in the task domain. The user of the ES need not be necessarily an expert in Artificial Intelligence. It explains how the ES has arrived at a particular recommendation. The explanation may appear in the following forms –



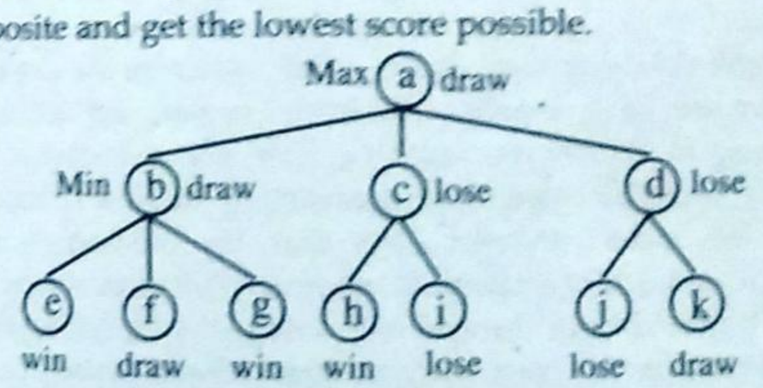
- Natural language displayed on screen.
- Verbal narrations in natural language.
- Listing of rule numbers displayed on the screen.

The user interface makes it easy to trace the credibility of the deductions.

9. **How mini-max algorithm is used in game search. For the following state space, show how mini-max algorithm finds path for the two players.** [2.5+2.5]




**Answer:** Mini-max is a kind of backtracking algorithm that is used in decision making and game theory to find the optimal move for a player, assuming that your opponent also plays optimally. It is widely used in two player turn-based games such as Tic-Tac-Toe, Backgammon, Chess, etc. In Mini-max the two players are called maximizer and minimizer. The maximizer tries to get the highest score possible while the minimizer tries to do the opposite and get the lowest score possible.




Max will move to b and min will respond by moving to f.

10. **How knowledge is represented using semantic networks? Illustrate with an example.** [5]

**Answer:** Knowledge representation (KR) is the study of how knowledge about the world can be represented and what kinds of reasoning can be done with that knowledge. Knowledge Representation is the method used to encode knowledge in Intelligent Systems.

Scanned by TapScanner

A semantic network, or frame network is a knowledge base that represents semantic relations between concepts in a network. This is often used as a form of knowledge representation. It is a directed or undirected graph consisting of vertices, which represent concepts, and edges, which represent semantic relations between concepts,[1] mapping or connecting semantic fields.
A semantic network is used when one has knowledge that is best understood as a set of concepts that are related to one another. Most semantic networks are cognitively based. They also consist of arcs and nodes which can be organized into a taxonomic hierarchy. Semantic networks contributed ideas of spreading activation, inheritance, and nodes as proto-objects.

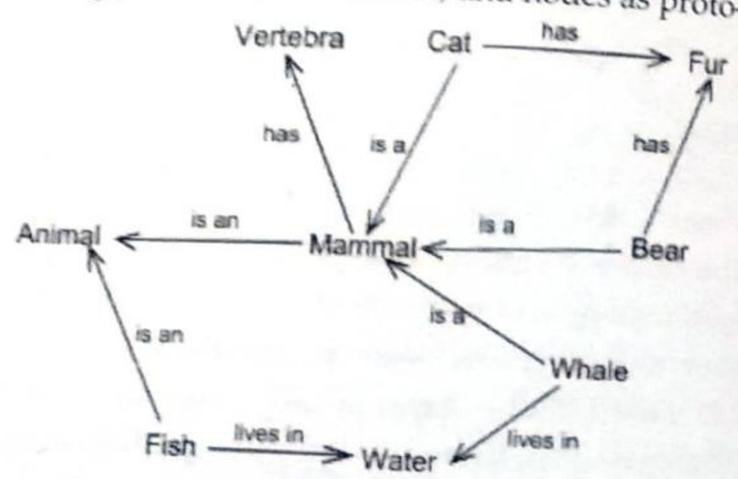



**Fig:** Semantic network

**11. What is supervised learning? Discuss how Naïve Bayes model works?**

**Answer:** Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically supervised learning is a learning in which we teach or train the machine using data which is well labeled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples (data) so that supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labeled data.
Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.
$Y = f(X)$
The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data. It is called supervised learning because the process of algorithm learning from the



training dataset can be thought of as a teacher supervising the learning process. We know the correct answers; the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.
Supervised learning problems can be further grouped into regression and classification problems.

- **Classification:** A classification problem is when the output variable is a category, such as "red" or "blue" or "disease" and "no disease".
- **Regression:** A regression problem is when the output variable is a real value, such as "dollars" or "weight".

Some common types of problems built on top of classification and regression include recommendation and time series prediction respectively. Some popular examples of supervised machine learning algorithms are:

- Linear regression for regression problems.
- Random forest for classification and regression problems.
- Support vector machines for classification problems.

**Naïve Bayes model**

Naive Bayes is a probabilistic machine learning algorithm based on the Bayes Theorem, used in a wide variety of classification tasks. The Bayes Rule provides the formula for the probability of Y given X. But, in real-world problems, you typically have multiple X variables. When the features are independent, we can extend the Bayes Rule to what is called Naive Bayes. It is called 'Naive' because of the naive assumption that the X's are independent of each other. Regardless of its name, it's a powerful formula. The representation for naive Bayes is probabilities.
Lists of probabilities are stored to file for a learned naive Bayes model. This includes:

- **Class Probabilities:** The probabilities of each class in the training dataset.
- **Conditional Probabilities:** The conditional probabilities of each input value given each class value.

**Calculating Class Probabilities**

The class probabilities are simply the frequency of instances that belong to each class divided by the total number of instances. For example in a binary classification the probability of an instance belonging to class 1 would be calculated as:

Scanned by TapScanner

P(class=1) = count(class=1) / (count(class=0) + count(class=1))
In the simplest case each class would have the probability of 0.5 or 50% for a binary classification problem with the same number of instances in each class.

**Calculating Conditional Probabilities**

The conditional probabilities are the frequency of each attribute value for a given class value divided by the frequency of instances with that class value. For example, if a "weather" attribute had the values "sunny" and "rainy" and the class attribute had the class values "go-out" and "stay-home", then the conditional probabilities of each weather value for each class value could be calculated as:

- P(weather=sunny | class=go-out) = count(instances with weather=sunny and class=go-out) / count(instances with class=go-out)
- P(weather=sunny | class=stay-home) = count(instances with weather=sunny and class=stay-home) / count(instances with class=stay-home)
- P(weather=rainy | class=go-out) = count(instances with weather=rainy and class=go-out) / count(instances with class=go-out)
- P(weather=rainy | class=stay-home) = count(instances with weather=rainy and class=stay-home) / count(instances with class=stay-home)

**12. Construct PEAS framework for following intelligent agents.**

**a. Internet Shopping Assistant**

**b. English Language Tutor**

**Answer**: PEAS frameworks for Internet Shopping Assistant are listed below:

- **Performance measure:** Price, quality, appropriateness, efficiency . . .
- **Environment:** Current and future Web sites, vendors, shippers . . .
- **Actuators:** Display to user, follow URL, fill in form . . .
- **Sensors:** Web pages (text, graphics, scripts. . .)

PEAS framework for English Lang Tutor are listed below:

- **Agent:** Interactive English t.
- **Performance measure:** Maximize student's score on test
- **Environment:** Set of students
- **Actuators:** Screen display (exercises, suggestions, corrections)
- **Sensors:** Keyboard



# Model Questions Sets For Practice

## MODEL SET 1

Course Title: Artificial Intelligence — Full Marks: 60
Course No: CSC261 — Pass Marks: 24
Level: B. Sc CSIT Second Year/ Fourth Semester — Time: 3 Hrs

### Section A

**Long Answer Questions**

**Attempt any two questions.** **[2*10=20]**

1. What is meant by admissible heuristic? Give example of admissible heuristic. What improvement is done in A* search than Greedy search? Prove that A* search gives us optimal solution if the heuristic function is admissible.
2. Describe role of domain expert, knowledge engineer and programmer in SDLC of expert system. Explain the features of expert system.
3. What is Bayesian network? Explain how Bayesian network represent and inference the uncertain knowledge.

### Section B

**Short Answer Questions**

**Attempt any eight questions.** **[8*5=40]**

4. What are basic components for production system? Explain it with diagram. Formulate condition Base Action Rules (if else...) & Trace it with suitable example.
5. What is heuristic information? Suppose that we run a greedy search algorithm with
   a. h(n)-g(n) and (ii) h(n)=g(n). What sort of search will the greedy search follow in each case?
6. Define the term Sate Space Representation. Explain with one example.
7. Define Toy Problem. What is working principle of 8 puzzle problem? Explain with example.
8. Anyone passing his history exams and winning the lottery is happy. But anyone who studies or is lucky can pass all his exams. John did not study but John is lucky. Anyone who is lucky wins the lottery. Is John happy? Solve the problem using resolution.

Scanned by TapScanner

9. Why neural network is more useful for machine learning? Explain different activation function of neural network.
10. Define alpha-beta pruning & explain it with suitable example. What are advantages of it for searching?
11. Define an artificial intelligence (AI). Describe your own criteria for computer program to be considered intelligent. What do you mean by Turing Test? Explain it.
12. What is standardizing part? Convert the following statements into FOPL
    - All over smart persons are stupid
    - Children's of all stupid persons are naughty
    - Hari is over smart
    - Prove that Ram is naughty by using backward chaining.

## MODEL SET 2

Course Title: Artificial Intelligence — Full Marks: 60
Course No: CSC261 — Pass Marks: 24
Level: B. Sc CSIT Second Year/ Fourth Semester — Time: 3 Hrs

### Section A

**Long Answer Questions**
**Attempt any two questions.**

1. What is the difference between symbolic and non-symbolic AI? Represent the following knowledge in semantic network
   Robin is bird
   Clyde is a Robin
   Clyde owns a nest from spring 2010 to fall 2010
2. What is machine learning? Explain the learning from Analogy and instance-based learning.
3. What is meant by complete and soundness property of inference algorithm? Consider the knowledge base given as below:
   $(P \leftrightarrow (P \vee Q)) \wedge \neg P$
   Prove that $\neg Q$ can be inferred from above KB by using resolution.

### Section B

**Short Answer Questions** **[8*5=40]**
**Attempt any eight questions.**

4. Why disjunctive normal form is required? Explain all the steps with example.



5. Differentiate between inference and reasoning. Why probabilistic reasoning is important in the AI? Explain with an example.
6. Justify that searching is one of the important part of AI. Explain in detail about depth first search and breadth first search techniques with an example.
7. Define learning. Why learning frame work is required? Explain about learning frame with block diagram and example.
8. What is back propagation? Explain all the steps involved in the back propagation with an example.
9. Define natural language processing. Explain the different issues involved in the natural language processing.
10. What is Bayes' theorem? Explain its applications.
11. What is heuristic information? Suppose that we run a greedy search algorithm with H(n)-g(n) and H(n)=g(n). What sort of search will the greedy search follow in each case?
12. What do you mean by knowledge representation? Explain the characteristics of representation.

## MODEL SET 3

Course Tit[2*10=20]al Intelligence — Full Marks: 60
Course No: CSC261 — Pass Marks: 24
Level: B. Sc CSIT Second Year/ Fourth Semester — Time: 3 Hrs

### Section A

**Long Answer Questions**
**Attempt any two questions.** **[2*10=20]**

1. Define the Model-based and Cased Based System. Discuss which system is suitable for the following problems.
   - Electronic Circuit Testing
   - Legal reasoning
   - Disease Recognition
2. What do you mean by forward chaining? Why it is required? Explain it with two practical examples.
3. Differentiate between natural language understanding (NLU) and natural language generating (NLG). Why we have to study natural language processing? Explain it.

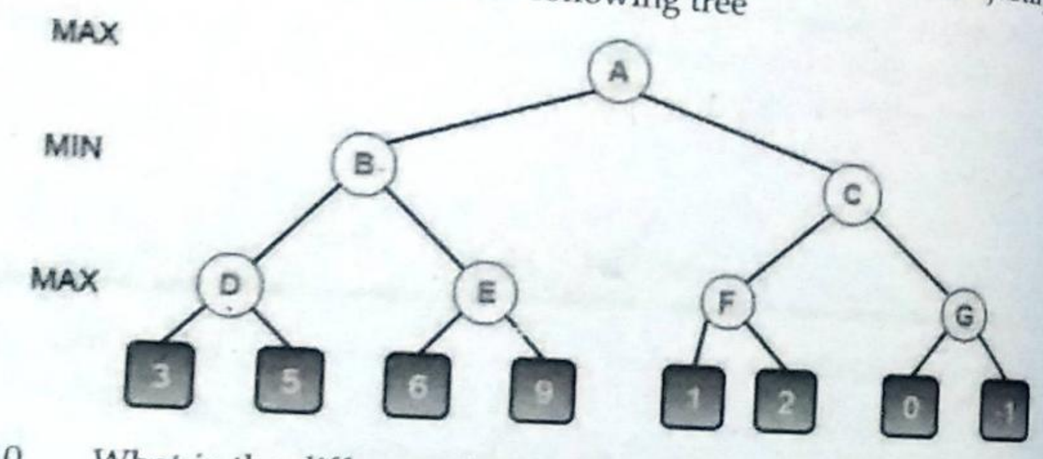
Scanned by TapScanner

## Section B

**Short Answer Questions**

**Attempt any eight questions.** [8*5=40]

4. What is a Neural Network? Explain any one type of neural network with practical example.
5. What do you mean by casual network? Explain it with practical application.
6. Justify that AI can't exist without searching. Explain in detail about any two types of informed search with practical examples.
7. What is a Constraint Satisfaction Problem? Create a CSP from the real world and suggest some measures to solve it.
8. What is a State Space Search? Give any example of a game which happens to be a problem of state space search and justify.
9. Perform mini-max on the following tree




10. What is the difference between symbolic and non-symbolic AI? Represent the following knowledge in semantic network.
    Robin is a bird.
    Clyde is a Robin.
    Clyde owns a nest from spring 2019 to fall 2019
11. Explain the steps of Natural Language Processing.
12. Given the following information for a database
    a) If x is on top of y, y supports x
    b) If x is above y and they are touching each other, x is on top of y
    c) A cup is above a book.
    d) A cup is touching a book.
    i. Translate statements a to d into clausal form.
    ii. Show that the predicate supports (book, cup) is tree using resolution



## MODEL SET 4

Course Title: Artificial Intelligence
Course No: CSC261
Level: B. Sc CSIT Second Year/ Fourth Semester

Full Marks: 60
Pass Marks: 24
Time: 3 Hrs

### Section A

**Long Answer Questions**

**Attempt any two questions.** [2*10=20]

1. Explain Alpha-Beta cutoff in Min-Max Algorithm with the help of an example.
2. Consider the following statement.
   "Mary likes all kinds of food. Pizza is a kind of food. Apple is a food. Anything anyone eats is a food. John does not eat chicken. Ana eats everything Mary eats."
   Convert these statements into predicate logic. Answer questions given below using Resolution algorithm.
   i. What food does Ana eat?
   ii. Is Mary like chicken?
   iii. Pizza is liked by Mary or not?
3. What are the difference between expert system and artificial neural networks?

### Section B

**Short Answer Questions**

**Attempt any eight questions.** [8*5=40]

4. Design Semantic Networks and frames for the banking system.
5. How to interpret the frames in Knowledge Representation?
6. Explain Alpha-Beta cutoff in Min-Max Algorithm with the help of an example.
7. Solve the problem for finding shortest path taking any example using A* algorithm?
8. Explain Production System. What is meaning of f(n), g(n), h(n).
9. What is the Turing Test for AI? Explain
10. Define Artificial Intelligence from the THINKING HUMANLY perspective. Give the PEAS description of the following task environment:
    a) Internet book-shopping agent.
    b) Automated Taxi
11. What do you mean by problem? Explain production system along with its architecture.
12. Give the mathematical model of Artificial Neural Network. Realize AND gate and OR gate using perception learning algorithm.

Scanned by TapScanner

## MODEL SET 5

Course Title: Artificial Intelligence
Course No: CSC261
Level: B. Sc CSIT Second Year/ Fourth Semester

Full Marks: 60
Pass Marks: 24
Time: 3 Hrs

### Section A

**Long Answer Questions**

**Attempt any two questions.**

1. What do you mean by Learning by simulating evolution? Give appropriate algorithm along with suitable example.
2. How can you design an expert system? Explain the role of knowledge engineer in the development of expert system.
3. Why searching is required? Give the idea of Simulated Annealing. How Simulated Annealing does removes the problem of Hill Climbing Search.

### Section B

**Short Answer Questions**

**Attempt any eight questions.** [8*5=40]

4. Discuss knowledge representation along with its use.
5. Explain Candidate Elimination Algorithm along with simple example trace.
6. Convert the following into Conjunctive Normal Form:
   $\neg\exists x\forall y\forall z\ ((P(y)\vee Q(z))\rightarrow(P(x)\vee Q(x)))$.
7. Explain Mini-Max search procedure with example.
8. Explain the uniformed search techniques with example.
9. What is Turing Test in AI? Criticize the performance of the Turing Test to measure the intelligence of the machine.
10. If we set the heuristic function h(n)=g(n) for both greedy as well as A* what will be effect in the algorithm? Explain.
11. Represent the following sentences into a semantic network.
    Birds are animals.
    Birds have feathers, fly and lay eggs.
    Albatross is a bird
    Donald is a bird
    Tracy is an albatross.
12. What is Bayesian network? Explain how Bayesian network represents and inference the uncertain knowledge.



## MODEL SET 6

Course Title: Artificial Intelligence
Course No: CSC261
Level: B. Sc CSIT Second Year/ Fourth Semester

Full Marks: 60
Pass Marks: 24
Time: 3 Hrs

### Section A

**Long Answer Questions** [2*10=20]

**Attempt any two questions.**

1. What are Conceptual graphs? Represent the following statements into conceptual graph.
   "King Ram marry Sita, the daughter of king Janak"
2. What is an export system? Explain the architecture and feature of rule based expert system.
3. Define learning. Why learning frame work is required? Explain about learning frame work with block diagram and example.

### Section B

**Short Answer Questions**

**Attempt any eight questions.** [8*5=40]

4. Define with suitable supporting statements and examples, "Artificial intelligence is the system that act like human"
5. For the following agents, determine what type of agent architecture is most appropriate (i.e. table lookup, simple reflex, foal-based or utility-based)
   a) Medical diagnosis system
   b) Satellite imagine analysis system
   c) Part-picking robot
   d) Refinery controller
6. Explain Candidate Elimination Algorithm along with simple example trace.
7. Discuss knowledge representation along with its use.
8. Why disjunctive normal form is required? Explain all the steps with examples.
9. Explain the steps of Natural language processing.
10. Convert the following sentence into predicate logic
    a) No dog bits a child of its owner
    b) No two adjacent countries have the same color
11. Define learning. Why learning frame work is required? Explain about learning frame work with block diagram and example.
12. Describe briefly the approaches of knowledge representation with example.

Scanned by TapScanner

## MODEL SET 7

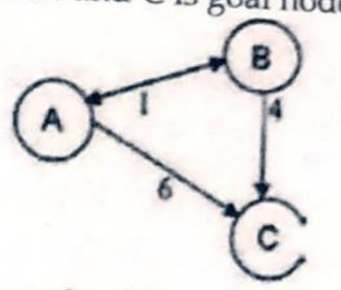
Course Title: Artificial Intelligence
Course No: CSC261
Level: B. Sc CSIT Second Year/ Fourth Semester

Full Marks: 60
Pass Marks: 24
Time: 3 Hrs

### Section A

**Long Answer Questions**
**Attempt any two questions.** [2*10=20]

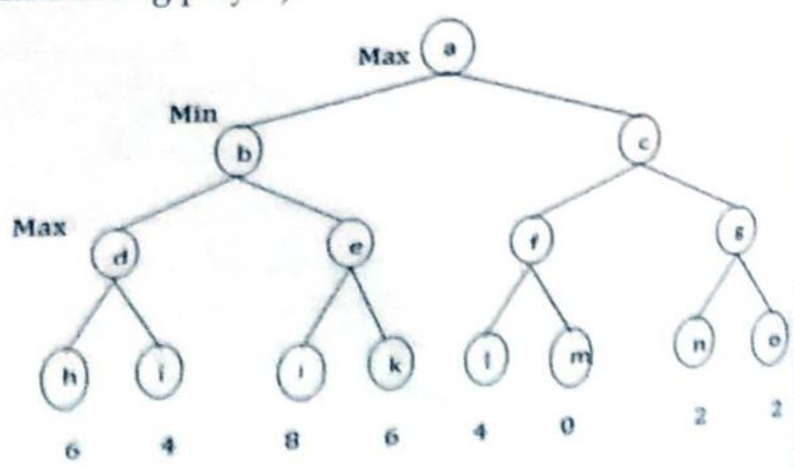
1. Consider the following graph, steps cost is given on the arrow. Assume that the successors of a state are generated in alphabetical order and that there is no repeated state checking. A is the starting node and C is goal node.




   a) Of the four algorithms breadth-first, depth first and iterative-deepening which find a solution in this case?
   b) Write sequence of node expanding by algorithm if finds solution.

2. Translate the following sentence into first order logic
   - Everyone's DNA is unique and is derived from their parent's DNA
   - No dog bites a child of its owner
   - Every gardener like the sun
   - All purple mushrooms are poisonous

3. What is the purpose of utility value in game search? Consider the following game tree (drown from the point of view of the Maximizing player):






Use the mini-max procedures and show what moves should be chosen by the two players. Use the alpha-beta pruning procedure and show what nodes would not need to be examined.

### Section B

**Short Answer Questions** [8*5=40]
**Attempt any eight questions.**

4. Differentiate between inference and reasoning. Why probabilistic reasoning is important in the AI? Explain with an example.
5. Define Learning. Why learning frame work is required? Explain about learning frame with block diagram and examples.
6. After your yearly checkup, the doctor has bad news and good news. The bad news is that you tested positive for a serious disease, and the test is 99% accurate (i.e. the probability of testing positive given that you have the disease is 0.99, as is the probability of testing negative if you don't have the disease). The good news is that this is a rare disease, striking only one in 10,000 people.
   a. Why is it good news that the disease is rare?
   b. What are the chances that you actually have the disease?
7. "System that think like humans" and "System that act like humans" are the part of artificial intelligence. Justify that statement with practical examples.
8. Consider the following sentence:
   [(food→party) V (drinks→party)] →[(food^drinks)→ party]
   a.) Convert the right hand and left hand sides of main implication into CNF.
   b.) Prove the validity of sentence using resolution.
9. What do you mean by forward chaining? Why it is required? Explain it with two practical examples.
10. Justify that AI can't exist without searching. Explain in detail about any two types of informed search with practical examples.
11. How do you convert to conjunctive normal form? Explain all the steps with example.
12. What is learning by induction? Explain inductive learning process with example.

Scanned by TapScanner

## MODEL SET 8

Course Title: Artificial Intelligence
Course No: CSC261
Level: B. Sc CSIT Second Year/ Fourth Semester

Full Marks: 60
Pass Marks: 24
Time: 3 Hrs

### Section A

**Long Answer Questions**
**Attempt any two questions.** [2*10=20]

1. Searching is an important part of AI, justify it. Explain any two types of blind search with suitable examples. How can you expand it to informed search?
2. What do you mean by reasoning in belief network? Explain it with suitable example.
3. Explain the different steps involved in the natural language processing (NLP) with block diagram and examples.

### Section B

**Short Answer Questions**
**Attempt any eight questions.** [8*5=40]

4. Derive the mathematical model of neural network. Explain any two types of neural networks with their block diagram and algorithms.
5. Why do we require expert system structure? Draw the block diagram and explain it with practical example.
6. What are conceptual graphs? Represent the following statements into conceptual graph.
   "Mr. Krishna marry Rupa , the daughter of Dr. Indra"
7. Define the model-based and case based system. Discuss which system is suitable for the following problems
   a) Electronic circuit testing
   b) Legal Reasoning
8. Consider the following statements:
9. Rabin likes only easy courses. Science courses are hard. All courses in the CSIT are easy. CSC101 is a CSIT course.
   a) Translate the sentences into predicate logic
   b) Convedrt your sentences into clausal normal form (CNF)
10. Write down the steps for resolution. Also prove that "You will enjoy" for the following facts using resolution graph
    a. If it is sunnay and warm day, you will enjoy
    b. If it is raining you will get wet
    c. It is a warm day
    d. It is raining
    e. It is sunny
11. Show steps to unify using Unification algorithm for (L1, L2).
12. Define with suitable supporting statements and examples, "Artificial intelligence is the system that act like humans".



## MODEL SET 9

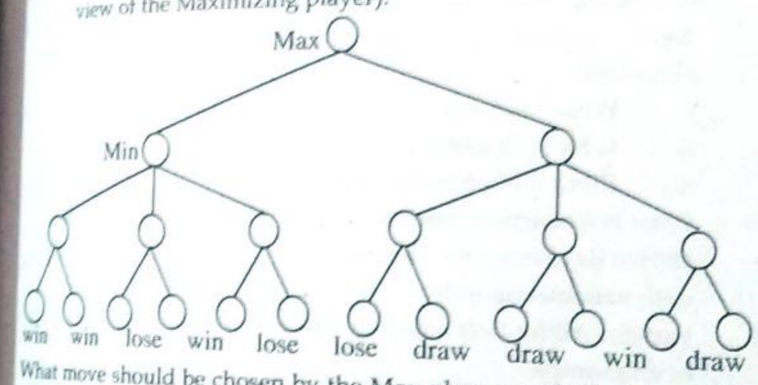
Course Title: Artificial Intelligence
Course No: CSC261
Level: B. Sc CSIT Second Year/ Fourth Semester

Full Marks: 60
Pass Marks: 24
Time: 3 Hrs

### Section A

**Long Answer Questions**
**Attempt any two questions.**

1. What is environment? Describe properties of environment with example. How plausible is human-like artificial intelligence?
2. Consider the following game tree (drown from the point of view of the Maximizing player):




What move should be chosen by the Max player, and what should be the response of the Min player, assuming that both are using the mini-max procedure?

3. For the 8-puzzle problem, assume the following initial state and goal state. Draw all possible states that can be reached from the initial state in one step. If the heuristic function is the number of pieces.in wrong place, calculate its value for each of the states listed previously, and decide which action to take at the initial state accordingly.

| | | |
|---|---|---|
| 2 | 4 | 3 |
| 6 | 5 | |
| 7 | 8 | 1 |

Initial State

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | | 5 |
| 6 | 7 | 8 |

Goal State

Scanned by TapScanner

## Section B

### Short Answer Questions

*Attempt any eight questions.* [8*5=40]

4. List the major disadvantages of artificial intelligence as compared to natural intelligence.
5. Define an intelligent agent. Why is it useful?
6. What are the difference between expert system and artificial neural networks?
7. Consider the following statement

   "Mary likes all kinds of food. Pizza is a kind of food. Apple is a food. Anyone eats is a food. John does not eat chicken. Ana eats everything Mary eats." Convert these statements into predicate logic. Answer questions given below using Resolution algorithm.

   i. What food does Ana eat?

   ii. Is Mary like chicken?

   iii. Pizza is liked by Mary or not?
8. What is a heuristic function? Explain Hill climbing problem.
9. Derive Bays theorem. Describe the importance of bays theorem with suitable example.
10. Explain Alpha-Beta cutoff in Min-Max algorithm with the help of an example.
11. Differentiate between simple reflexive and model based agents with example.
12. Covert the following sentences into Predicate calculus:

    Robert is an American. All Americans who sells weapons are criminals. All missiles are weapons. Robert sells weapons. Prove that Robert is criminal.

...

Scanned by TapScanner