NOTES ON
Discrete Mathematics
MCA I Year/ I Semester
(2023-24)

By

Dr. K.V.S.Sudhakar
(PRINCIPAL, ASSOCIATE PROFESSOR)

DEPARTMENT OF MCA
RAMNATH GULJARILAL KEDIA COLLEGE OF COMMERCE
3-1-336, OPP NEW CHADERGHAT BRIDGE,
Kachiguda Station Road,
Esamia Bazar, Hyderabad 500 027, Telangana,India.

(PCC101) DISCRETE MATHEMATICS

Course Objectives

1. Use mathematically correct terminology and notation.
2. Construct correct direct and indirect proofs.
3. Use division into cases in a proof.
4. Use counterexamples.
5. Apply logical reasoning to solve a variety of problems

Course Outcomes

1. For a given logic sentence express it in terms of predicates, quantifiers, and logical connectives
2. For a given a problem, derive the solution using deductive logic and prove the solution based on logical inference
3. For a given a mathematical problem, classify its algebraic structure
4. Evaluate Boolean functions and simplify expressions using the properties of Boolean algebra
5. Develop the given problem as graph networks and solve with techniques of graph theory.

Suggested Readings:

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw – Hill
2. Susanna S. Epp, Discrete Mathematics with Applications,4th edition, Wadsworth Publishing Co. Inc
3. C L Liu and D P Mohapatra, Elements of Discrete Mathematics A Computer Oriented Approach, 3rd Edition by, Tata McGraw – Hill.
4. J.P. Tremblay and R. Manohar, “Discrete Mathematical Structure and it’s Application to Computer Science”, TMG Edition, Tata Mcgraw-Hill
5. Norman L. Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press. Schaum’s Outlines Series, Seymour Lipschutz, Marc Lipson.

1

NUMBER SYSTEM

Unit Structure

1.0 OBJECTIVES

After going through this unit you will have knowledge :

- Decimal number system, binary number system, octal number system and hexadecimal number system.
- Conversion of numbers from one system to other system.
- Binary arithmetic - addition, subtraction, multiplication and division.

1.1 INTRODUCTION

This unit mainly deals with the representation of various number systems used in the computer system. The smallest piece of data recognized and used by a computer is ‘bit’ or binary digit. The binary system consists of two digits O and 1. In the system, ‘O’ can be represented by electricity ‘off’ and ‘1’ by electricity being ‘on’. All the numbers, letters and special characters that are entered in a computer are internally represented in binary numbers. The computer only understands ‘O’ and ‘1’. It executes all operation using machine language which consists of only ‘O’ and ‘1’. Hence it is important to understand and study in detail the representation of binary numbers.

1.2 THE DECIMAL NUMBER SYSTEM

The number systems are based on an ordered set of number of digits. The total number of digits used in a number system is called ‘base’ or ‘radix’ of the number system. The decimal number system used 10 digits - 0, 1, 2, ….. 9 and hence its base is 10. The base of binary number system is 2, base of octal number system is 8 and hexadecimal system has base 16. The decimal number system is the most popular system used not only by scientists and engineers but also by the common man. It is a positional number system as the value of any number depends on the position of digits.

e.g. Consider number 638.

Its representation is 638 = 600 + 30 + 8

i.e. $638_{10} = 6 \times 10^2 + 3 \times 10^1 + 8 \times 10^0$

Starting from left most digit, 6 is in the hundreds position. It is called t he Most Significant Digit (MSD). 3is in the tens position. The last digit 8 is in the unit position and is the least significant digit (LSD).

The example shows that the value of each position is a power of base 10. The power can be either positive, negative or 0.

Power of 10 and its value:

$10^0 = 1$	$10^{-0} = \frac{1}{1} = 1$
$10^1 = 10$	$10^{-1} = 1/10 = 0.1$
$10^2 = 100$	$10^{-2} = 1/100 = 0.01$
$10^3 = 1000$	$10^{-3} = 1/1000 = 0.001$
$10^4 = 10000$	$10^{-4} = 1/10000 = 0.0001$
$10^5 = 100000$	$10^{-5} = 1/100000 = 0.00001$

e.g. The number 3254.78 will be represented as

$$3254.78 = 3\times10^3 + 2\times10^2 + 5\times10^1 + 4\times10^0 + 7\times10^{-1} + 8\times10^{-2}$$

1.3 BINARY NUMBER SYSTEM

1.3.1 What is binary number system?

A computer stores numbers, letters and characters in a coded form which is a series of string of Os and 1s. The binary number system consists of only O and 1. The digits in binary system are called ‘bits’. A group of 4 bits is called nibble and a group of 8 bits is called a byte. A group of 16 bits is known as ‘word’ and a group of 32 bits is called a ‘double word’. This number system has base 2. Computers are designed to handle only binary numbers because computer circuits have to handle only two binary digits which simplifies the design of the circuits, reduces cost and improves reliability.

In binary system, the value of each digit is based on 2 and powers of 2.

1.3.2 Decimal to binary system conversion:

Step 1) Divide the number by 2, the remainder is either O or 1.

2) Place the remainder to the right of number.

3) Subsequently divide the partial quotient by 2 and again place the remainder to the right of the partial quotient.

4) Repeat the steps till we get the partial quotient O.

5) The binary number is equal to the remainders arranged so that the first remainder is the LSD and last remainder is MSD.

i.e. in order from bottom to top.

Example 1. Convert 65_{10} into binary.

Successive dividers	Original Number & partial quotients	Remainders
2	65	1
2	32	0
2	16	0
2	8	0
2	4	0
2	2	0
2	1	1

0 → STOP

$65_{10} = 1000001_2$

Example 2. Convert 78_{10} to binary.

Successive Dividers	Original Number & Partial quotients	Remainders
2	78	0
2	39	1
2	19	1
2	9	1
2	4	0
2	2	0
2	1	1
	0	

$\therefore\ 78_{10} = 1001110_2$

1.3.3 Binary to decimal conversion :

To convert binary number to its equivalent decimal number, multiply the extreme right digit by 2^0, the second digit from right by 2^1, the third digit from right by 2^2 and so on till we reach the left most digit. Then add all these products. The sum is the decimal equivalent of the binary number.

Example 1. Convert 1001110_2 to decimal number.

$$\begin{aligned} 1001110_2 &= 1\times 2^6 + 0\times 2^5 + 0\times 2^4 + 1\times 2^3 + \times 2^2 + 1\times 2^1 + 0\times 2^0 \\ &= 64 + 0 + 0 + 8 + 4 + 2 + 0 \\ &= 78_{10} \end{aligned}$$

Example 2 : Convert 110101_2 to decimal.

$$110101_2 = 1\times 2^5 + 1\times 2^4 + 0\times 2^3 + 1\times 2^2 + 0\times 2^1 \times 1\times 2^0$$

$= 32 + 16 + 6 + 4 + 0 + 1$

$= 53_{10}$

1.3.4 Decimal to binary fraction conversion :

Step 1) Multiply the decimal fraction by 2.

2) After multiplication, if a number equal or greater than 1 is obtained then place 1 on the right of the partial product. If the product is less than 1, place 0 to the right of the partial product.

3) The partial product obtained in step 2 is multiplied by 2. The process is repeated till the partial product is 0 or the resulting binary fraction is to the required places of binary point

4) The ones and zeroes in the order obtained are equal to the binary fraction.

5) The order is from top to bottom.

Example 1 : $0.625_{10} = ?_2$

Successive Multiplier		Decimal fraction & Partial product		
2	×	0.625	= 1.25	1
2	×	0.25	= 0.5	0
2	×	0.5	= 1.0	1

$\therefore 0.625_{10} = 0.101_2$

Example 2 : $0.86_{10} = ?_2$

2	×	0.86	= 1.72	1
2	×	0.72	= 1.44	1
2	×	0.44	= 0.88	0
2	×	0.88	= 1.76	1

$\therefore 0.86_{10} = 0.1101_2$

Example 3 : $50.7_{10} = ?_2$

In the last two examples we have converted for only decimal numbers. In this example we will convert for both 50 and .7 & then combine them.

We should be careful while writing order for 50 (the order is from top to bottom) and for .7 (the order is from bottom to top).

Successive Dividers	Original Number & Partial quotients	Remainders		Successive Multiplier	Decimal Fraction & Partial Product
2	50	0		$2\times 0.7=1.4$	1
2	25	1		$2\times 0.4=0.8$	0
2	12	0		$2\times 0.8=1.6$	1
2	6	0		$2\times 0.6=1.2$	1
2	3	1	↑		
2	1	1			
	0				

Therefore

$50_{10}=110010_2$ $\qquad$ $0.7_{10}=0.1011_2$

$\therefore\ 50.7_{10}=110010.1011_2$

1.3.5 Binary to decimal fraction conversion :

To convert a binary fraction to decimal fraction multiply the first bit after binary point by 2^{-1}, the second by 2^{-2}, third by 2^{-3} and so on. Add all these products to get the decimal equivalent.

Example 1. $0.1111_2=?_{10}$

$$\begin{aligned}0.\underrightarrow{1111}&=1\times 2^{-1}+1\times 2^{-2}+1\times 2^{-3}+1\times 2^{-4}\\&=1\times\frac{1}{2}+1\times\frac{1}{4}+1\times\frac{1}{8}+1\times\frac{1}{16}\\&=0.5+0.25+0.125+0.0625\\&=0.9375_{10}\end{aligned}$$

Example 2. $111011.101_2=?_{10}$

$$\begin{aligned}111011&=1\times 2^5+1\times 2^4+1\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0\\&=32+16+8+0+2+1\\&=59\end{aligned}$$

$$\begin{aligned}0.101&=1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}\\&=1/2+0+1/8\\&=0.5+0+0.125\\&=0.625\end{aligned}$$

$111011.101_2=59.625_{10}$

Check your progress:

1) $96_{10}=?_2$	Ans: 1100000_2
2) $1000001_2=?_{10}$	Ans: 65_{10}
3) $11010_2=?_{10}$	Ans: 26_{10}

4) $154_{10} = ?_2$ Ans: 10011010_2

5) $1001010.1010001_2 = ?_{10}$ Ans: 74.6328125_{10}

6) $101001.1101_2 = ?_{10}$ Ans: 41.8125_{10}

7) $678.67_{10} = ?_2$ Ans: 1010100110.1010_2

8) $59.625_{10} = ?_2$ Ans: 111011.101_2

1.4 OCTAL NUMBER SYSTEM

1.4.1 What is octal number system?

Octal system was used to deal with the long strings of 1s and 0s in binary. It is a base 8 system using the digits 0, 1, 2, 3, 4, 5, 6 and 7. Thus each digit of an octal number can have only values 0 to 7.

The digit position in an octal number have weights as follows-

$$8^4\ 8^3\ 8^2\ 8^1\ 8^0 \underset{\substack{\uparrow \\ \text{Octal point}}}{\bullet} 8^{-1}\ 8^{-2}\ 8^{-3}\ 8^{-4}$$

The largest octal digit is 7. After 7, the next digit is taken to be 10.

Octal	
0	
1	
2	
3	
4	
5	
6	
7	$7_8 + 1_8 = 10_8$
10	
11	
⋮	
16	
17	$17_8 + 1_8 = 20_8$
20	
⋮	
27	$27_8 + 1_8 = 30_8$
30	
⋮	
	& so on

1.4.2 Decimal to Octal conversion:

Steps:

1) Divide the decimal number by 8.
2) Place the remainder to the right of original number.

3) Subsequently divide the partial quotient by 8 and place the remainder to the right of partial quotient.

4) Repeat the above steps till we get partial quotient 0.

5) The octal number is equal to the remainder arranged so that first remainder is LSD and last remainder is MSD of the octal number (i.e. from down to up)

Example 1. $119_{10} = ?_8$

Successive dividers	**Original Number & partial quotients**	**Remainders**
8	119	7
8	14	6
8	1	1
	0	

$\therefore\ 119_{10} = 167_8$

2. $2536_{10} = ?_8$

Successive dividers	**Original Number & partial quotients**	**Remainders**
8	2536	0
8	317	5
8	39	7
8	4	4
	0	

$\therefore\ 2536_{10} = 4750_8$

1.4.3 Decimal to Octal fraction Conversion :

Steps:

1) Multiply the decimal fraction by 8.

2) Write the integer to the right of the product i.e. if 0.6 x 8 = 4.8 then place 4 to the right of the product.

3) The partial product is again multiplied by 8 and the integer is placed to the right of the product.

4) Repeat the process till the partial products is seen or till the required place of octal point.

Example 1 $0.96_{10} = ?_8$

0.96 × 8	=	7.68	7
0.68 × 8	=	5.44	5
0.44 × 8	=	3.52	3
0.52 × 8	=	4.16	4
0.16 × 8	=	1.28	1
0.28 × 8	=	2.24	
0.96_{10}	=	0.75341_8	

2. $0.5625_{10} = ?_8$

$0.5625 \times 8 = 4.5$ 4

$0.5 \times 8 = 4.0$ 4

$\therefore 0.5625_{10} = 0.44_8$

3. $73.52_{10} = ?_8$

Successive dividers	Original number & partial quotient	Reminders
8	73	1
8	9	1
8	1	1
	0	

$8 \times 0.52 = 4.16$ 4

$8 \times 0.16 = 1.28$ 1

$8 \times 0.28 = 2.24$ 2

$8 \times 0.24 = 1.92$ 1

$0.52_{10} = 0.4121_8$

$73_{10} = 111_8$

$\therefore 0.73.52_{10} = 111.4121_8$

1.4.4 Octal Decimal Conversion :

To convert a whole octal number to its decimal equivalent, the extreme right hand digit is multiplied by 8^0, the second digit from right is multiplied by 8^1, the third from right is multiplied by 8^2, and so on. Then all their products are added to get the required decimal number.

To convert octal fraction to decimal fraction-

Multiply first digit after octal point by 8^{-1}, second digit after octal point by 8^{-2}, third digit after octal point by 8^{-3} & so on. Add all their products to get the required decimal fraction.

Example 1. $56_8 = ?_{10}$

$56 = 5 \times 8^1 + 5 \times 8^0 = 40 + 6 = 46$

$\therefore 56_8 = 46_{10}$

2) $642_8 = 6 \times 8^2 + 4 \times 8^1 + 4 \times 8^0$

$= 6 \times 64 + 4 \times 8 + 2$

$= 384 + 32 + 2 \quad = \quad 418_{10}$

3) $0.563_8 = ?_{10}$

$0.563 = 5 \text{ x } 8^{-1} + 6 \text{ x } 8^{-2} + 3 \text{ x } 8^{-3}$

$$= \frac{5}{8} + \frac{6}{64} + \frac{3}{512}$$

$= 0.625 + 0.09375 + 0.0058\ \ 59375$

$= 0.724609375$

$0.563_8 = 0.724609375_{10}$

4) $111.4121_8 = \quad ?_{10}$

$111 \ = 1 \text{ x } 8^2 + 1 \text{ x } 8^1 + 1 \text{ x } 8^0 = 64 + 8 + 1 = 73$

$0.\ 4121_8 \quad = 4 \text{ x } 8^{-1} + 1 \text{ x } 8^{-2} + 2 \text{ x } 8^{-3} + 1 \text{ x } 8^{-4}$

$$= \frac{4}{8} + \frac{1}{8^2} + \frac{2}{8^3} + \frac{1}{8^4}$$

$= 0.5 + 0.015625 + 0.00390625 + 0.00024414$

$= 0.51977539_{10}$

$111.4121_8 = 73.51977539_{10}$

Check your progress :

1) $364_8 = ?_{10}$	Ans : 244_{10}
2) $72_8 = ?_{10}$	Ans : 58_{10}
3) $119_{10} = ?_8$	Ans : 167_8
4) $634.640625_{10} = ?_8$	Ans : 1172.51_8
5) $0.96_{10} = ?_8$	Ans : 0.753412_8
6) $454_8 = ?_{10}$	Ans : 300_{10}
7) $0.135_8 = ?_{10}$	Ans : 0.1816406_{10}

1.4.5 Octal to Binary Conversion

The conversion of octal to binary is done by converting each octal digit to its 3-bit binary equivalent

Example 1 56_8

Successive Dividers	**Original Number & Partial quotients**	**Remainders**		**Successive Dividers**	**Original Number & Partial quotients**	**Remainders**
2	5	1		2	6	0
2	2	0		2	3	1
2	1	1		2	1	
2	0				0	
2			↑			
2						

$5_8 \ = 101_2$ $6_8 \ = 110_2$

$56_8 \ = 101110_2$

OR

We can convert 56_8 to decimal number and then convert the decimal number to its binary equivalent.

Check : $56_8 = 46_{10} = 101110_2$

Example 2 $0.216_8 = ?_2$

Successive Dividers	Original Number & Partial quotients	Remainders		Successive Dividers	Original Number & Partial quotients	Remainders	Successive Dividers	Original Number & Partial quotients	Remainders
2	2	0	↑	2	1	1	2	6	0
2	1	1			0		2	3	1
	0						2	1	
								0	

010 001 110

Note if it is less than 3 - bit add 0 in the left to make 3 bit.

$\therefore$ $0.216_8 = 0.010001110_2$

3) $576.12_8 = ?_2$

Successive Dividers	Original Number & Partial quotients	Remainders		Successive Dividers	Original Number & Partial quotients	Remainders	Successive Dividers	Original Number & Partial quotients	Remainders
2	5	1		2	7	1	2	6	0
2	2	0	↑	2	3	1	2	3	1
2	1	1		2	1	1	2	1	1
	0				0			0	

101 111 110

$576_8 = 101111110_2$

0.12_8

Successive Dividers	Original Number & Partial quotients	Remainders		Successive Dividers	Original Number & Partial quotients	Remainders
2	1	1		2	2	0
	0			2	1	1
					0	

001 010

$0.12_8 = 0.001010_2$

$576.12_8 = 101111110.001010_2$

1.4.6 Binary to Octal Conversion:

Steps:

1) Make a group of 3 bits, starting from binary point
2) For whole numbers make group of three from right to left (from binary point)
3) For fractional part, move left to right from binary point.
4) In case of one or two bits left, add zeroes to make a group of three.
5) Replace each group of 3 bits by equivalent octal numbers.

Example 1 $1101011_2 = ?_8$

$\underline{1}\,\underline{101}\,\underline{011}$

001 101 011 (Complete group of 3 bits by adding zeroes)

001 = $0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$ = 0 + 0 + 1 = 1

101 = $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$ = 4 + 0 + 1 = 5

011 = $0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$ = 0 + 2 + 1 = 3

$\therefore 1101011_2 = 153_8$

2) $0.010101_2 = ?_8$

$\underline{0.010}\,\underline{101}$

010 = $0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$ = 0 + 2 + 0 = 2

101 = $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$ = 4 + 0 + 1 = 5

$\therefore \quad 0.010101_2 = 0.25_8$

3) $1101.11101_2 = ?$

$\underline{1}\ \underline{101}.\underline{111}\ \underline{01}$

$\underline{001}\ \underline{101}.\underline{111}\ \underline{010}$

This will give 15.72_8 (check)

$\therefore\ 1101.11101_2 = 15.72_8$

Check your progress:

1) $637_8 = ?_2$ Ans : 110011111_2
2) $256_8 = ?_2$ Ans : 10101110_2
3) $56.34_8 = ?_2$ Ans : 101110.011100_2
4) $1011.1011_2 = ?_8$ Ans : 13.54_8
5) $0.1101_2 = ?_8$ Ans : 0.64_8

1.5 HEXADECIMAL NUMBER SYSTEM

1.5.1 What is hexadecimal number system?

Hexadecimal number system is a system with base 16. Thus it is a system which has 16 possible digit symbols. As we are familiar with only 10 digits - 0 to 9, the hexadecimal system uses letters A to F to represent the remaining numbers 10 to 15.

Thus the 16 digit symbols are
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F.

Also four binary digits are grouped together to represent each digit in hexadecimal number system.

1.5.2 Decimal to Hexadecimal Conversion -

Steps-

1) Divide the number by 16
2) Place the remainder to the right of original number.
3) Subsequently divide the partial quotient by 16 and place the remainder to the right of partial quotient.
4) Repeat the above steps till we get quotient 0
5) Then the hexadecimal number is equal to the remainders arranged from the last remainder to the first remainder (ie. down to up)

Example 1 : $10761_{10} = ?_{16}$

Successive Dividers	Original Number & Partial quotients	Remainders
16	10761	9
16	672	0
16	42	10
16	2	2
	0	

$10761_{10} = 2\ (10)\ 09$
$= 2\ A\ 09_{16}$

2) $6747_{10} = ?_{16}$

Successive Dividers	Original Number & Partial quotients	Remainders
16	6747	11
16	421	5
16	26	10
16	1	1
	0	

$6747_{10} = 1\ (10)\ 5\ (11)$
$= 1\ A\ 5B_{16}$

Decimal to hexadecimal fraction conversion-
Steps :

1) Multiply decimal fraction by 16

2) Write the integer to the right of product.

3) The partial product is again multiplied by 16 and integer is placed to the right of product.

4) Repeat the process till partial product is seen or till the required place of hexadecimal point.

Example 1 $0.256_{10} = ?_{16}$

0.256 × 16 = 4.096	4
0.096 × 16 = 1.536	1
0.536 × 16 = 8.576	8
0.576 × 16 = 9.216	9

2) $97.236_{10} = ?_{16}$

16	97	1
16	6	6
	0	

$97_{10} = 61_{16}$

0.236 × 16 = 3.776	3
0.776 × 16 = 12.416	12
0.416 × 16 = 6.656	6
0.656 × 16 = 10.496	10

∴ $0.236_{10} = 0.3(12)\ 6\ (10) = 0.3C6A_{16}$

∴ $97.236_{10} = 61.3C6A_{16}$

1.5.3 Hexadecimal to Decimal conversion-

1) To convert a whole hexadecimal number to its decimal equivalent, the extreme right digit is multiplied by 16^0, the second from right by 16^1, the third digit from right by 16^2 and so on. Add all their products to get the required decimal number.

2) To convert hexadecimal fraction to decimal fraction, multiply the first digit after hexadecimal point by 16^{-1}, the second digit from point by 16^{-2} and so on. Add all these products to get the equivalent decimal number.

Example 1 $1A5E_{16} = ?_{10}$

1) 1A5E

= $1 \times 16^3 + A \times 16^2 + 5 \times 16^1 + E \times 16^0$

= $4096 + 10 \times 256 + 80 + 14 \times 1$

= 6750_{10}

2) $AB7_{16}$

= $A \times 16^2 + B \times 16^1 + 7 \times 16^0$

= $10 \times 256 + 11 \times 16 + 7$

= 2743_{10}

3) $61.3C6A_{16} = ?_{10}$

61 = $6 \times 16^1 + 1 \times 16^0 = 96 + 1 = 97$

$$
\begin{aligned}
0.3C6A &= 3 \times 16^{-1} + C \times 16^{-2} + 6 \times 16^{-3} + A \times 16^{-4} \\
&= \frac{3}{16} + \frac{12}{256} + \frac{6}{4096} + \frac{10}{65536} \\
&= 0.1875 + 0.0469 + 0.0015 + 0.0002 \\
&= 0.2361
\end{aligned}
$$

$\therefore\ 61.3C6A_{16} = 97.2361_{10}$

Check your progress :

1)	$3A9_{16} = ?_{10}$	Ans. 937_{10}
2)	$7551_{10} = ?_{16}$	Ans. $1D7F_{16}$
3)	3370.75_{10} = ?16	Ans. $D2A.6_{16}$
4)	$0.3942_{10} = ?_{16}$	Ans. 0.64EA
5)	$0.48_{16} = ?_{10}$	Ans. 0.28125_{16}

1.5.4 Hexadecimal to Binary conversion

The conversion from hexadecimal to binary is performed by converting each hexadecimal digit to its 4-bit binary equivalent.

Example 1. $4C3F_{16} = ?_2$

C = 12

F = 15

2	4	0
2	2	0
2	1	1
	0	

100

2	12	0
2	6	0
2	3	1
2	1	1
	0	

1100

2	3	1
2	1	1
	0	

11

2	15	1
2	7	1
2	3	1
2	1	1
	0	

1111

4 bit is (0100) (1100) 4 bit is (0011) (1111)

$\therefore \quad 4C3F_{16} = 0100110000111111_2$

2) AB.CD_{16} = $?_2$

	A = 10				B = 11				C = 12				D = 13		
2	10	0		2	11	1		2	12	0		2	13	1	
2	5	1		2	5	1		2	6	0		2	6	0	
2	2	0	↑	2	2	0	↑	2	3	1		2	3	1	↑
2	1	1		2	1	1		2	1	1		2	1	1	
	0				0				0				0		
	(1010)				(1011)				(1100)				(1101)		

∴ AB.CD_{16} = 10101011.11001101_2

1.5.5 Binary to hexadecimal Conversion -

1) Group the binary bits into fours starting from binary point.
2) For whole number, make group of four form right to left from binary point.
3) For fractional part, make group of four from left to right from binary point.
4) In case you are left with only one or two or three bits, add zero or zeroes at appropriate places.
5) Replace each group by equivalent hexadecimal numbers (by multiplying by powers 2^3 to 2^0)

Example 1 : 1101001100_2 = $?_{16}$

11 0100 1100

(0011) (0100) (1100)

0011 = $0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 2+1 = 3$

0100 = $0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 4$

1100 = $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 12 = C$

∴ 1101001100_2 = $34C_{16}$

2) 110101.1111101_2 = $?_{16}$

11 0101.1111 101

(0011) (0101).(1111)(1010)

0011 = $0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 3$

0101 = $0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$

1111 = $1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 15 = F$

1010 = $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 10 = A$

∴ 110101.1111101_2 = $35.FA_{16}$

Check your progress :

1) $F2E_{16} = ?_2$ Ans. 111100101110_2
2) $6BC_{16} = ?_2$ Ans. 011010111100_2
3) $10110110111_2 = ?_{16}$ Ans. $5B7_{16}$
4) $1111110_2 = ?_{16}$ Ans. $7E_{16}$
5) $0.0101011_2 = \ ?_2$ Ans. $0.5B_{16}$
6) $0.2\ D6_{16} = ?_2$ Ans. 0.111011010110_2

1.5.6 Hexadecimal to Octal numbers-

Steps-

1) First convert each digit in number to its binary equivalent (by dividing by 2) and write it in group of 4 bits.
2) Then make group of 3 bits each from right to left.
3) Again convert it into binary equivalent (by multiplying by powers 2^2 to 2^0)

Octal to hexadecimal numbers-

Steps-

1) Convert each digit in binary equivalent (by dividing by 2) and write in group of 3 bits each.
2) Then take group of 4 bits each from right to left.
3) Again convert it into binary equivalent (by multiplying by powers 2^3 to 2^0)

Example 1 : $3F2_{16} = ?_8$

F = 15

2	3	1		2	15	1		2	2	0	
2	1	1	↑	2	7	1		2	1	1	↑
	0			2	3	1	↑		0		
				2	1	1					
					0						

(0011) (1111) (0010)

$\underline{001}\,\underline{111}\,\underline{110}\,\underline{010}$

001	=	$0 \times 2^2 + 0 \times 2^1 + 1 \times 2^6 = 1$
111	=	$1 \text{ x } 2^2 + 1 \text{ x } 2^1 + 1 \text{ x } 2^0 = 7$
110	=	$1 \text{ x } 2^2 + 1 \text{ x } 2^1 + 0 \text{ x } 2^0 = 6$
010	=	$0 \text{ x } 2^2 + 1 \text{ x } 2^1 + 0 \text{ x } 2^0 = 2$
	∴	$(3F2)_{16} = 1762_8$

2) $1527_8 = ?_{16}$

2	1	1		2	5	1		2	2	0		2	7	1	
	0		↑	2	2	0	↑	2	1	1	↑	2	3	1	↑
				2	1	1			0			2	1	1	
					0								0		

(001) (101) (010) (111)

$\underline{0011}\,\underline{0101}\,\underline{0111}$

0011	=	$0 \text{ x } 2^3 + 0 \text{ x } 2^2 + 1 \text{ x } 2^1 + 1 \text{ x } 2^0 = 3$
0101	=	$0 \text{ x } 2^3 + 1 \text{ x } 2^2 + 0 \text{ x } 2^1 + 1 \text{ x } 2^0 = 5$
0111	=	$0 \text{ x } 2^3 + 1 \text{ x } 2^2 + 1 \text{ x } 2^1 + 1 \text{ x } 2^0 = 7$
	∴	$1527_8 = 357_{16}$

3) $47.43_8 = ?_{16}$

2	4	0		2	7	1		2	4	0		2	3	1
2	2	0		2	3	1		2	2	0		2	1	1
2	1	1		2	1	1		2	1	1			0	
	0				0				0					
	(100)				(111)				(100)				(011)	

100111.100011

10 0111.100 011

0010 0111.1000 1100

0010 = $0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 2$

0111 = $0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 4 + 2 + 1 = 7$

1000 = $1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 8$

1100 = $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 8 + 4 = 12 = C$

$\therefore$ $47.43_8 = 27.8C_{16}$

Remember : When we are converting

Hexadecimal to octal number first convert each digit in group of 4 bits then make group of 3 bits from right to left.

But when we are converting octal to Hexadecimal number first convert each digit in group of 36 bits then take group of 4 bits each from right to left.

Check your prograss -

1)	$5A3_{16} = ?_8$	Ans : 2643_8
2)	$4753_8 = ?_{16}$	Ans : $9EB_{16}$

1.6 BINARY ARITHMETIC

1.6.1 Binary Addition:

Rules :
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 i.e. take it as 0 with a carry of 1.

Example 1 Add 110101 and 101111

```
    1 1 0 1 0 1
+   1 0 1 1 1 1
  -------------
  1 1 0 0 0 0 0
```

```
2)     10 110              3)     1010
    +   1 101                     1000
       100011                  +  0110
                                  0111
                                   11111
```

1.6.2 Binary Subtraction

```
Rules :   0 - 0 = 0
          1 - 1 = 0
          1 - 0 = 1
          0 - 1 = 1       and     borrow 1
```

```
                 ***                          *   *
eg.    1)       110101               2)      10110
             +  101111                     -   1101
                000110                         1001
       * columns are borrowed from
                    **                          *  ***
       3)       11011.0              4)      1101110
             -    1001.1                   -    10111
                10001.1                      1010111


       5)       1000101              6)      110011
             -    101100                   -   10110
                   11001                       11101
```

1.6.3 **Multiplication:**

```
Rules :   0 x 0 = 0
          0 x 1 = 0
          1 x 0 = 0
          1 x 1 = 1
```

```
eg.    1)       10110            2)       111
             x    1101                  x  101
                10110                      111
               00000x                     000x
             + 10110xx                  + 111xx
              10110xxx                  100011
            100011110

       3)       1111
             x    111
                1111
               1111x
             + 1111xx
              1101001
```

Note: Check multiplication by checking their equivalent decimal multiplication.

1.6.4 Division:

Division for binary numbers can be carried out by following same rules as those of decimal system.

eg. 1) Divide 100011 by 101

```
101)100011(111
   - 101
      0111
      - 101
        0101
      -   101
            0
Ans: 111
```

2) Divide 11110 by 110

```
110)11110(101
   -  110
        11
      -   0
        110
      - 110
          0

Ans: 101
```

3)

```
1110)1000110(101
     -  1110
         00111
           -   0
           1110
        - 1110
              0
Ans: 101
```

Check your progress:

1)	1100 + 1011	Ans:	10111
2)	11101 + 10011	Ans:	110000
3)	1100 + 1010 + 1101 + 0111	Ans:	101010
4)	1110 - 1011	Ans:	0011

5) 11110011 - 1110001 Ans: 10000010

6) 11010 x 1011 Ans: 100011110

7) 0010111 x 0000011 Ans: 1000101

8) 00101010 $\div$ 00000110 Ans: 111

9) 00100101 - 00010001 Ans: 10100

1.7 SUMMARY

Computer uses only binary digits O and I. A binary digit is called a bit. There are two states in a bit - O and 1. In this unit we have seen four number systems. Decimal system has base 10, binary system - base 2, octal system - base 8 and hexadecimal system has base 16. Three binary digits correspond to one octal digit and four binary digits translate into one hexadecimal digit. The following table shows the four systems.

Binary	Octal	Hexadecimal	Decimal
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	A	10
1011	13	B	11
1100	14	C	12
1101	15	D	13
1110	16	E	14
1111	17	F	15

In binary arithmetic, addition is simply the bitwise XOR operation with carry and multiplication is simply logical AND operation. Subtraction is equivalent to adding a negative number and division is equivalent to multiplying by the inverse.

1.8 UNIT END EXERCISES

1) $(76)_{10} = (?)_2$ Ans: $(1001100)_2$

2) $11000.0011_2 = ?_{10}$ Ans: 24.1875

3) Convert the following binary numbers into decimal numbers.

i)	11001.0101	Ans:	25.3125
ii)	1101.11	Ans:	13.75
iii)	1001.101	Ans:	9.625
iv)	1011001	Ans:	89
v)	0.1011	Ans:	0.6875
vi)	0.0101	Ans:	0.3125

4) Convert decimal into binary numbers.

i)	97	Ans:	1100001
ii)	154	Ans:	10011010
iii)	17.71875	Ans:	10001.1011
iv)	74.635	Ans:	1001010.1010001
v)	43	Ans:	101011

5) Convert octal to decimal number

i)	5264	Ans:	2740
ii)	642	Ans:	418
iii)	704	Ans:	452
iv)	134	Ans:	92
v)	1075.6256	Ans:	573.79248

6) Convert decimal to octal numbers -

i)	8_{10}	Ans:	10
ii)	2749	Ans:	5275
iii)	9	Ans:	11
iv)	3965	Ans:	7575
v)	460	Ans:	714
vi)	201	Ans:	311

7) Convert octal to binary numbers -

i)	435	Ans:	100011101
ii)	13.54	Ans:	1011.1011
iii)	134	Ans:	1011100
iv)	576.216	Ans:	101111110.010001110
v)	56.34	Ans:	101110.011100

8) Convert binary to octal numbers -

i)	110111101	Ans:	675
ii)	11000110	Ans:	306
iii)	1111000	Ans:	170
iv)	1101.11101	Ans:	15.72
v)	0.1101	Ans:	0.64

9) Convert hexadecimal to decimal numbers -

i) $1F95_{16}$ Ans: 8085_{10}

ii) $475C_{16}$ Ans: 18268_{10}

iii) $0.D2F_{16}$ Ans: 0.82397_{10}

iv) D6C1 Ans: 54977_{10}

10) Convert decimal to hexadecimal numbers -

i) 3370 Ans: D2A

ii) 70 Ans: 46

iii) 0.62 Ans: 0.9EB851

iv) 10761 Ans: 2A09

v) 6747 Ans: 1A5B

11) Convert hexadecimal to binary numbers -

i) $59C_{16}$ Ans: 010110011100_2

ii) 6D.3A Ans: 1101101.00111010_2

iii) 6BC Ans: 11010111100_2

iv) 43CF Ans: 100001111001111_2

12) Convert binary to hexadecimal numbers -

i) 11011110_2 Ans: DE_{16}

ii) 110000110_2 Ans: 186_{16}

iii) 0.011011_2 Ans: $0.6C_{16}$

13) Convert octal to hexadecimal numbers -

i) 46.57 Ans: 26.BC

ii) 134 Ans: 5C

14) Convert hexadecimal to octal numbers -

i) 4B Ans: 113

ii) 5B.3A Ans: 133.164

15) Perform the following Binary Arithmetic -

i) 10101 + 1110 Ans: 100011

ii) 101 + 1010 Ans: 1111

iii) 1101 + 1111 Ans: 11100

iv) $1011_2 - 0110_2$ Ans: 0101

v) $1101 \div 1001$ Ans: approx 1.0111

vi) $1101_2 \times 1010$ Ans: 11111010

1.9 REFERENCES

1) Computer fundamentals - B.Ram (4th edition)

   New Age International Publishers

2) Fundamentals of Computers - V. Rajaraman (4th edition)

   Prentice Hall - India

3) Computer fundamentals (3rd edition)

   Pradeep Sinha & Priti Sinha - BPB publications

4) Digital computer fundamentals - 6th edition

   - Thomas Bartee (Tata McGraw Hili)

5) Computer fundamentals - D.P. Nagpal (S. Chand)

6) Computer fundamentals - A.B.Patil, D. Ravichandran (for MSBTE) - Tata McGraw Hill

❖❖❖❖

2

PROPOSITIONS AND LOGICAL OPERATIONS

Unit Structure :

2.0 OBJECTIVES :

After going through this unit, you will be able to :

- Define statement & logical operations.
- Define & to use the laws of Logic.
- Describe the logical equivalence and implications.
- Define arguments & valid arguments.
- Test the validity of argument using rules of logic.
- Give proof by truth tables.
- Give proof by mathematical Induction.

2.1 INTRODUCTION :

Mathematics is an exact science. Every statement in Mathematics must be precise. Also there can't be Mathematics without proofs and each proof needs proper reasoning. Proper reasoning involves logic. The dictionary meaning of 'Logic' is the science of reasoning. The rules of logic gives precise meaning to mathematic statements. These rules are used to distinguished between valid & invalid mathematical arguments.

In addition to its importance in mathematical reasoning, logic has numerous applications in computer science to verify the correctness of programs & to prove the theorems in natural & physical sciences to draw conclusion from experiments, in social sciences & in our daily lives to solve a multitude of problems.

The area of logic that deals with propositions is called the propositional calculus or propositional logic. The mathematical approach to logic was first discussed by British mathematician George Boole; hence the mathematical logic is also called as Boolean logic.

In this chapter we will discuss a few basic ideas.

2.2 PROPOSITIONS (OR STATEMENTS)

A proposition (or a statement) is a declarative sentence that is either true or false, but not both.

A proposition (or a statement) is a declarative sentence which is either true or false but not both.

Imperative, exclamatory, interrogative or open statements are not statements in logic. Mathematical identities are considered to be statements.

Example 1 : For Example consider, the following sentences.

i) The earth is round.

ii) $4 + 3 = 7$

iii) London is in Denmark

iv) Do your homework

v) Where are you going?

vi) $2 + 4 = 8$

vii) $15 < 4$

viii) The square of 4 is 18.

ix) $x+1=2$

x) May God Bless you!

All of them are propositions except iv), v), ix) & x) sentences i), ii) are true, where as iii), iv), vii) & viii) are false.

Sentence iv) is command hence not proposition. Is a question so not a statement. ix) Is a declarative sentence but not a statement, since it is true or false depending on the value of x. x) is a exclamatory sentence and so it is not statement.

Mathematical identities are considered to be statements.

Statements which are imperative, exclamatory, interrogative or open are not statements in logic.

<u>Compound statements :</u>

Many propositions are composites that is, composed of subpropositions and various connectives discussed subsequently. Such composite propositions are called compound propositions.

A proposition is said to be primitive if it can not be broken down into simpler propositions, that is, if it is not composite.

Example 2 : Consider, for example following sentences.

(a) "The sum is shining today and it is cold"

(b) "Juilee is intelligent or studies every night."

Also the propositions in Example 1 are primitive propositions.

2.3 LOGICAL OPERATIONS OR LOGICAL CONNECTIVES :

The phrases or words which combine simple statements are called logical connectives.

For example, 'and', 'or', 'note', 'if……then', 'either…….or' etc….

In the following table we list some possible connectives, their symbols & the nature of the compound statement formed by them.

Sr. No.	Connective	Symbol	Compound statement
1	AND	$\wedge$	Conjuction
2	OR	$\vee$	Disjunction
3	NOT	$\neg$	Negation
4	If……..then	$\rightarrow$	Conditional or implication
5	If and only if (iff)	$\leftrightarrow$	Biconditional or equivalence

Now we shall study each of basic logical connectives in details.

Basic Logical Connectives :

2.3.1 Conjunction (AND) :

If two statements are combined by the word “and” to form a compound proposition (statement) is called the conjunction of the original proposition.

Symbolically, if P & Q are two simple statements, then ‘ $P \wedge Q$ ’ denotes the conjunction of P and Q and is read as ‘P and Q.

Since, $P \wedge Q$ is a proposition it has a truth value and this truth value depends only on the truth values of P and Q.

Specifically, if P & Q are true then $P \wedge Q$ is true; otherwise $P \wedge Q$ is false.

The truth table for conjunction is as follows.

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

Example 3 :

Let P : Monsoon is very good this year.
Q : The rivers are rising.

then
$P \wedge Q$: Monsoon is very good this year and rivers are rising.

2.3.2 Disjunction (OR) :

Any two statements can be connected by the word ‘or’ to form a compound statement called disjunction.

Symbolically, if P and Q are two simple statements, then $P \vee Q$ denotes the disjunction of P and Q and read as $'P \text{ or } Q'$.

The truth value of $P \vee Q$ depends only on the truth values of P and Q. specifically if P and Q are false then $P \vee Q$ is false otherwise $P \vee Q$ is true.

The truth table for disjunction is as follows.

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

Example 4 :

P : Paris is in France
Q : $2+3=6$
then $P \vee Q$: Paris is in France or 2 + 3 = 6.
Here, $P \vee Q$ is True since P is true & Q is False.

Thus, the disjunction $P \vee Q$ is false only when P and Q are both false.

2.3.3 Negation (NOT)

Given any proposition P, another proposition, called negation of P, can be formed by writing "It is not the case that…….. or". "It is false that……" before P or, if possible, by inserting in P the word "not".

Symbolically $\neg P$ or $\sim P$ read "not P" denotes the negation of P. the truth value of $\neg P$ depends on the truth value of P.

If P is true then $\neg P$ is false and if P is false then $\neg P$ is true. The truth table for Negation is as follows :

P	$\neg P$
T	F
F	T

Example 5 :

Let P : 6 is a factor of 12.
Then Q = $\neg P$: 4 is not a factor of 12.
Here P is true & $\neg P$ is false.

2.3.4 Conditional or Implication : (If……then)

If two statements are combined by using the logical connective 'if….then' then the resulting statement is called a conditional statement.

If P and Q are two statements forming the implication "if P then Q" then we denotes this implication $P \to Q$.

In the implication $P \to Q$,

P is called antecedent or hypothesis

Q is called consequent or conclusion.

The statement $P \to Q$ is true in all cases except when P is true and Q is false.

The truth table for implication is as follows.

P	Q	$P \to Q$
T	T	T
T	F	F
F	T	T
F	F	T

Since conditional statement play an essential role in mathematical reasoning a variety of terminology is used to express $P \to Q$.

i) If P then Q

ii) P implies Q

iii) P only if Q

iv) Q if P

v) P is sufficient condition for Q

vi) Q when P

vii) Q is necessary for P

viii) Q follows from P

ix) if P, Q

x) Q unless $\neg P$

Converse, Inverse and Contrapositive of a conditional statement : We can form some new conditional statements starting with a conditional statements related conditional statements that occur so often that they have special names → converse, contrapositive & Inverse. Starting with a conditional statement $P \to Q$ that occur so often that they have special names.

1. **Converse :** If $P \to Q$ is an implication then $Q \to P$ is called the converse of $P \to Q$.

2. **Contrapositive :** If $P \to Q$ is an implication then the implication $\neg Q \to \neg P$ is called it's contrapositive.

3. **Inverse :** If $P \to Q$ is an implication then $\neg P \to \neg Q$ is called its inverse.

Example 6 :

Let P : You are good in Mathematics.
Q : You are good in Logic.

Then, $P \to Q$: If you are good in Mathematics then you are good in Logic.

1) Converse : $(Q \to P)$
   If you are good in Logic then you are good in Mathematics.

2) Contrapositive : $\neg Q \to \neg P$
   If you are not good in Logic then you are not good in Mathematics.

3) Inverse : $(\neg P \to \neg Q)$
   If you are not good in Mathematics then you are not good in Logic.

2.3.5 Biconditional Statement : Let P and Q be propositions. The biconditional statement $P \leftrightarrow Q$ is the proposition "P if and only if Q". The biconditional statement is true when P and Q have same truth values and is false otherwise.

Biconditional statements are also called bi-implications. It is also read as p is necessary and sufficient condition for Q.

The truth table for biconditional statement is as follows.

P	Q	$P \leftrightarrow Q$
T	T	T
T	F	F
F	T	F
F	F	T

Example 7 : Let P : You can take the flight.
Q : You buy a ticket.
Then $P \leftrightarrow Q$ is the statement.
“You can take the flight iff you buy a ticket”.

Precedence of Logical Operators :

We can construct compound propositions using the negation operator and the logical operators defined so far. We will generally use parentheses to specify the order in which logical operators in a compound proposition are to be applied. In order to avoid an excessive number of parantheses.

We sometimes adopt an order of precedence for the logical connectives. The following table displays the precedence levels of the logical operators.

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

2.4 LOGICAL EQUIVALANCE :

Compound propositions that have the same truth values in all possible cases are called logically equivalent.

Definition : The compound propositions P and Q are called logically equivalent if $P \leftrightarrow Q$ is a tautology. The notation $P \equiv Q$ denotes that P and Q are logically equivalent.

Some equivalance are useful for deducing other equivalance. The following table shows some important equivalance.

2.4.1 Logical Identities or Laws of Logic :

Name	Equivalance
1. Identity Laws	$P \wedge T \equiv P$ $P \vee F \equiv P$
2. Domination Laws	$P \vee T \equiv T$ $P \wedge F \equiv F$
3. Double Negation	$\neg(\neg P) \equiv P$
4. Idempotent Laws	$P \vee P \equiv P$ $P \wedge P \equiv P$
5. Commutative Laws	$P \vee Q \equiv Q \vee P$ $P \wedge Q \equiv Q \wedge P$
6. Associative Laws	$(P \vee Q) \vee R \equiv P \vee (Q \vee R)$ $(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$

7. Distributive Laws	$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$ $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
8. De Morgan's Laws	$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$ $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
9. Absorption Laws	$P \vee (P \wedge Q) \equiv P$ $P \wedge (P \vee Q) \equiv P$
10. Negation Laws (Inverse / Complement)	$P \vee \neg P \equiv T$ $P \wedge \neg P \equiv F$
11. Equivalance Law	$P \leftrightarrow Q \equiv (P \to Q) \wedge (Q \to P)$
12. Implication Law	$P \to Q \equiv \neg P \vee Q$
13. Biconditional Property	$P \leftrightarrow Q \equiv (P \wedge Q) \vee (\neg P \wedge \neg Q)$
14. Contrapositive of Conditional statement	$P \to Q \equiv \neg Q \to \neg P$

Note that while taking negation of compound statement ‘every’ or ‘All’ is interchanged by ‘some’ & ‘there exists’ is interchanged by ‘at least one’ & vice versa.

Example 8 : If P : “This book is good.”

Q : “This book is costly.”

Write the following statements in symbolic form.

a) This book is good & costly.

b) This book is not good but costly.

c) This book is cheap but good.

d) This book is neither good nor costly.

e) If this book is good then it is costly.

Answers :

a) $P \wedge Q$

b) $\neg P \wedge Q$

c) $\neg Q \wedge P$

d) $\neg P \wedge \neg Q$

e) $P \to Q$

2.4.2 Functionally complete set of Connectives :

We know that there are five logical connectives $\neg, \vee, \wedge, \to$ and $\leftrightarrow$. But some of these can be expressed in terms of the other & we get a smaller set of connectives.

The set containing minimum number of connectives which are sufficient to express any logical formula in symbolic form is called as the functionally complete set of connectives.

There are following two functionally complete set of connectives.

(1) $\{\neg, \vee\}$ is complete set connectives.

Here, the $\wedge$ can be expressed using $\neg$ & $\vee$.

$\therefore P \wedge Q \equiv \neg \neg (P \wedge Q)$

$\equiv \neg (\neg P \vee \neg Q)$

The $\rightarrow$ can be expressed in terms of $\neg, \vee$.

$\therefore P \rightarrow Q \equiv \neg P \vee Q$

The $\leftrightarrow$ can be expressed in terms of $\neg, \vee$

$\therefore P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$

$\equiv (\neg P \vee Q) \wedge (\neg Q \vee P)$

$\equiv \neg [\neg (\neg P \vee Q) \vee \neg (\neg Q \vee P)]$

$\therefore \{\neg, \vee\}$ is a functionally complete set of connectives.

Similarly, you can prove that $\therefore \{\neg, \wedge\}$ is complete set of connectives.

2.5 LOGICAL IMPLICATIONS:

A proposition P (p, q, ……..) is said to logically imply a proposition Q (p, q, …….) written,

P (p, q, ……..) $\Rightarrow$ Q (p, q, …….) if Q (p, q, …….) is true whenever P (p, q, …….) is true.

Example 9 : $P \Rightarrow (P \vee Q)$

Solution :

Consider the truth table for this

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

Observe that if P is true (T) in rows 1 and 2 then $P \vee Q$ is also true (T) . $\therefore P \Rightarrow P \vee Q$.

If Q (p, q, …….) is true whenever P (p, q, …….) is true then the argument. $P(p, q, ...) \vdash Q(p, q, ...)$ is valid and conversely.

i.e. the argument $P \vdash Q$ is valid iff the conditional statement $P \to Q$ is always true, i.e. a tautology.

2.5.1 Logical Equivalence Involving Implications :

Let P & Q be two statements.

The following table displays some useful equivalences for implications involving conditional and biconditional statements.

Sr. No.	Logical Equivalence involving implications
1	$P \to Q \equiv \neg P \vee Q$
2	$P \to Q \equiv \neg Q \to \neg P$
3	$P \vee Q \equiv \neg P \to Q$
4	$P \wedge Q \equiv \neg(P \to \neg Q)$
5	$\neg(P \to Q) \equiv P \wedge \neg Q$
6	$(P \to Q) \wedge (P \to r) \equiv P \to (Q \wedge r)$
7	$(P \to r) \wedge (Q \to r) \equiv (P \vee Q) \to r$
8	$(P \to Q) \vee (P \to r) \equiv P \to (Q \vee r)$
9	$(P \to r) \vee (Q \to r) \equiv (P \wedge Q) r$
10	$P \leftrightarrow Q \equiv (P \to Q) \wedge (Q \to P)$
11	$P \leftrightarrow Q \equiv \neg P \leftrightarrow \neg Q$
12	$P \leftrightarrow Q \equiv (P \wedge Q) \vee (\neg P \wedge \neg Q)$
13	$\neg(P \leftrightarrow Q) \equiv P \leftrightarrow \neg Q$

All these identities can be proved by using truth tables.

2.6 NORMAL FORM AND TRUTH TABLES :

2.6.1 Well Formed Formulas : (wff)

A compound statement obtained from statement letters by using one or more connectives is called a statement pattern or statement form. thus, if P, Q, R, ……. are the statements (which can be treated as variables) then any statement involving these statements and the logical connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ is a statement form or a well formed formula or statement pattern.

Definition : A propositional variable is a symbol representing any proposition. Note that a propositional variable is not a proposition but can be replaced by a proposition.

Any statement involving a propositional variable and logical connectives is a well formed formula.

Note : A wff is not a proposition but we substitute the proposition in place of propositional variable, we get a proposition.

E.g. $\neg(P \vee Q) \wedge (\neg Q \wedge R) \rightarrow Q, \neg(P \rightarrow Q)$ etc.

2.6.1 (a) Truth table for a Well Formed Formula :

If we replace the propositional variables in a formula α by propositions, we get a proposition involving connectives. If α involves n propositional constants, we get 2n possible combination of truth variables of proposition replacing the variables.

Example 10 : Obtain truth value for $\alpha = (P \rightarrow Q) \wedge (Q \rightarrow P)$.

Solution : The truth table for the given well formed formula is given below.

P	Q	$P \rightarrow Q$	$Q \rightarrow P$	α
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

2.6.1 (b) Tautology :

A tautology or universally true formula is a well formed formula, whose truth value is T for all possible assignments of truth values to the propositional variables.

Example 11 : Consider $P \vee \neg P$, the truth table is as follows.

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

$\because P \vee \neg P$ always takes value T for all possible truth value of P, it is a tautology.

2.6.1 (c) Contradiction :

A contradiction or (absurdity) is a well formed formula whose truth value is false (F) for all possible assignments of truth values to the propositional variables.

Thus, in short a compound statement that is always false is a contradiction.

Example 12 : Consider the truth table for $P \wedge \neg P$.

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

$\therefore P \wedge \neg P$ always takes value F for all possible truth values of P, it is a contradiction.

2.6.1. (d) Contingency :

A well formed formula which is neither a tautology nor a contradiction is called a contingency.

Thus, contingency is a statement pattern which is either true or false depending on the truth values of its component statement.

Example 13 : Show that $\neg(p \vee q)$ and $\neg p \wedge \neg q$ are logically equivalent.

Solution : The truth tables for these compound proposition is as follows.

1	2	3	4	5	6	7	8
P	Q	$\neg P$	$\neg Q$	$P \vee Q$	$\neg(P \vee Q)$	$\neg P \wedge \neg Q$	$6 \leftrightarrow 7$
T	T	F	F	T	F	F	T
T	F	F	T	T	F	F	T
F	T	T	F	T	F	F	T
F	F	T	T	F	T	T	T

We cab observe that the truth values of $\neg(p \vee q)$ and $\neg p \wedge \neg q$ agree for all possible combinations of the truth values of p and q.

It follows that $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$ is a tautology, therefore the given compound propositions are logically equivalent.

Example 14 : Show that $p \rightarrow q$ and $\neg p \vee q$ are logically equivalent.

Solution : The truth tables for these compound proposition as follows.

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

As the truth values of $p \rightarrow q$ and $\neg p \vee q$ are logically equivalent.

Example 15 : Determine whether each of the following form is a tautology or a contradiction or neither :

i) $(P \wedge Q) \rightarrow (P \vee Q)$

ii) $(P \vee Q) \wedge (\neg P \wedge \neg Q)$

iii) $(\neg P \wedge \neg Q) \rightarrow (P \rightarrow Q)$

iv) $(P \rightarrow Q) \wedge (P \wedge \neg Q)$

v) $[P \wedge (P \rightarrow \neg Q) \rightarrow Q]$

Solution:

i) The truth table for $(p \wedge q) \rightarrow (p \vee q)$

P	q	$p \wedge q$	$p \vee q$	$(p \wedge q) \rightarrow (p \vee q)$
T	T	T	T	T
T	F	F	T	T
F	T	F	T	T
F	F	F	F	T

$\because$ All the entries in the last column are 'T'.

$\therefore (p \wedge q) \rightarrow (p \vee q)$ is a tautology.

ii) The truth table for $(p \vee q) \wedge (\neg p \wedge \neg q)$ is

1	2	3	4	5	6	
p	q	$p \vee q$	$\neg p$	$\neg q$	$\neg P \wedge \neg q$	$3 \wedge 6$
T	T	T	F	F	F	F
T	F	T	F	T	F	F
F	T	T	T	F	F	F
F	F	F	T	T	T	F

The entries in the last column are 'F'. Hence $(p \vee q) \wedge (\neg p \wedge \neg q)$ is contradiction.

iii) The truth table is as follows.

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

$\because$ All entries in last column are 'T'.
$\therefore$ $(\neg p \wedge \neg q) \to (p \to q)$ is a tautology.

iv) The truth table is as follows.

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

All the entries in the last column are 'F'. Hence it is contradiction.

v) The truth table for $\left[p \wedge (p \rightarrow \neg q) \rightarrow q\right]$

p	q	$\neg q$	$p \rightarrow \neg q$	$p \wedge (p \rightarrow \neg q)$	$\left[p \wedge (p \rightarrow \neg q) \rightarrow q\right]$
T	T	F	F	F	T
T	F	T	T	T	F
F	T	F	T	F	T
F	F	T	T	F	T

The last entries are neither all ‘T’ nor all ‘F’.

$\therefore$ $\left[p \wedge (p \rightarrow \neg q) \rightarrow q\right]$ is a neither tautology nor contradiction. It is a contingency.

2.6.2 Normal Form of a well formed formula :

One of the main problem in logic is to determine whether the given statement is a tautology or a contradiction. One method to determine it is the method of truth tables. Other method is to reduce the statement form to, called normal form.

If P & Q are two propositional variables we get various well formed formula.

The number of distinct truth values for formulas in P and Q is 2^4. Thus there are only 16 distinct formulae & any formula in P & Q is equivalent to one of these formulas.

Here we give a method of reducing a given formula to an equivalent form called a ‘normal form’. We use ‘sum’ for disjunction, ‘product’ for conjunction and ‘literal’ either for P or for $\neg P$, where P is any propositional variable.

Elementary Sum & Elementary Product :

An elementary sum is a sum of literals. An elementary product is a product of literals.

e.g. $P \vee \neg Q$, $P \vee \neg R$ are elementary sum $P \wedge \neg Q$, $\neg P \wedge Q$ are elementary products.

Disjunctive Normal Form (DNF) :

A formula is in disjunctive normal form if it is a sum of elementary products.

e.g. $P \vee (\neg Q \wedge R)$, $P \vee (Q \wedge R)$

A conjunction of statement variables and their negations are called as fundamental conjunctions. It is also called min term.

e.g. P, $\neg P$, $P \wedge \neg Q$

Construction to obtain a Disjunctive Normal Form of a given formula

The following procedure is used to obtain a disjunctive normal form.

1. Eliminate $\rightarrow$ and $\leftrightarrow$ using logical identifies.
2. Use De-Morgans laws to eliminate $\neg$ before sums or products.

   The resulting formula has $\neg$ only before propositional variables i.e. it involves sum, product and literals.
3. Apply distributive laws repeatedly to eliminate product of sums.

   The resulting formula will be sum of products of literals i.e. sum of elementary products.

Example 16 :

Obtain a disjunctive normal form of

1. $P \wedge (P \rightarrow Q)$
2. $(P \rightarrow Q) \wedge (\neg P \wedge Q)$
3. $(P \wedge \neg(Q \wedge R)) \vee (P \rightarrow Q)$

Answer :

1) Consider, $P \wedge (P \rightarrow Q)$

$\equiv P \wedge (\neg P \vee Q)$ (Implication law)

$(P \wedge \neg P) \vee (P \wedge Q)$ (Distributive law)

This is a disjunctive normal form of the given formula.

2) Using Implication law $P \rightarrow Q \equiv \neg P \vee Q$

$\therefore (P \rightarrow Q) \wedge (\neg P \wedge Q)$

$\equiv (\neg P \vee Q) \wedge (\neg P \wedge Q)$ Implication law

$\equiv (\neg P \wedge Q) \wedge (\neg P \vee Q)$ Commutative law

$\equiv (\neg P \wedge Q \wedge \neg P) \vee (\neg P \wedge Q \wedge Q)$ Distributive law

$\equiv (\neg P \wedge \neg P \wedge Q) \vee (\neg P \wedge Q \wedge Q)$ Associative law

$\equiv (\neg P \wedge Q) \vee (\neg P \wedge Q)$

This is required disjunctive normal form

3) $\left(p\wedge\neg\left(Q\wedge R\right)\right)\vee\left(P\to Q\right)$

$\equiv\left(P\wedge\neg\left(Q\wedge R\right)\right)\vee\left(\neg P\vee Q\right)$ Implication law

$\equiv\left(P\wedge\left(\neg Q\vee\neg R\right)\right)\vee\left(\neg P\vee Q\right)$ De-Morgans law

$\equiv\left(\left(P\wedge\neg Q\right)\vee\left(P\wedge\neg R\right)\right)\vee\left(\neg P\vee Q\right)$ Distributive law

$\equiv\left(P\wedge\neg Q\right)\vee\left(P\wedge\neg R\right)\vee\left(\neg P\vee Q\right)$ Associative law

This is the disjunctive normal form of the given formula.

- Note that for the same formula we may get different disjunctive normal forms. So we introduce one or more normal forms called the principle disjunctive normal form or sum of products canonical form in the next definition. The advantage of constructing principle disjunctive normal form is that for a given formula principle disjunctive normal form is unique.

- Two forms are said to be equivalent iff their principle disjunctive normal forms consider.

*** Min term :**

A min term in n propositional variables P_1, P_2, ……, P_n is $Q_1\wedge Q_2\wedge.......\wedge Q_n$ where each Q_i is either P_i or $\neg$ P_i.

e.g.

The min terms in P_1 & P_2 are $P_1 \wedge P_2$, $P_1 \wedge \neg P_2$, $\neg P_1 \wedge P_2$, $\neg P_1 \wedge \neg P_2$,

In general the number of min terms in n propositional variables is 2^n.

2.6.3 Principle Disjunctive Normal Form :

A formula α is in principle disjunctive normal form if α is a sum of min terms.

Steps to Construct Principle Disjunctive Normal Form of a given Formula : -

1. First obtain the disjunctive normal form for given formula.
2. Drop elementary products, which are contradiction such as $(P\wedge\neg P)$
3. If P_i & $\neg$ P_i are not present in an elementary product α, replace α by $\left(\alpha\wedge P_i\right)\vee\left(\alpha\wedge\neg P_i\right)$

4. Use the above step until all elementary products are reduced to sum of min terms.

   Use idempotent laws to avoid repetition of min terms.

Example 17 :

Obtain the canonical sum of product form i.e. principle disjunctive normal form of

1. $\alpha \equiv P \vee (\neg P \wedge \neg Q \wedge R)$
2. α whose truth table is given below

Row No.	P	Q	R	α
1	T	T	T	T
2	T	T	F	F
3	T	F	T	F
4	T	F	F	T
5	F	T	T	T
6	F	T	F	F
7	F	F	T	F
8	F	F	F	T

Answer :

1) α is already in disjunctive normal form. There are no contradictions. So we have to introduce missing - variables.

$\neg P \wedge \neg Q \wedge R$ in α is a min - term.

As $P \equiv (P \wedge Q) \vee (P \wedge \neg Q)$

$$\therefore P \equiv (P \wedge Q \wedge R) \vee (P \wedge Q \wedge \neg R) \vee (P \wedge \neg Q \wedge R) \vee (P \wedge \neg Q \wedge \neg R)$$

Therefore the canonical sum of products form of α is

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

2) For given α, we have T in column corresponding to rows 1, 4, 5 and 8. The min terms corresponding to these rows are $P \wedge Q \wedge R, P \wedge \neg Q \wedge \neg R \neg P \wedge Q \wedge R$ and $\neg P \wedge \neg Q \wedge \neg R$

$\therefore$ The principle disjunctive normal form of α is

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

Fundamental disjunction (Max term)

A disjunction of statement variables and (or) their negations are called as fundamental disjunctions. It is also called **max term**.

e.g. $P, \neg P, \neg P \wedge Q, P \wedge Q, P \vee \neg P \vee Q$

Conjunctive Normal Form : -

A statement form which consists of a conjunction of a fundamental disjunction is called a conjunctive normal form.

e.g. $P \wedge Q, (P \vee Q) \wedge \neg P$

If α is in disjunctive normal form then $\neg\alpha$ is in conjunctive normal form.

Maxterm

A max term in n propositional variables $P_1, P_2 ....P_n$ is $Q_1 \vee Q_2 \vee .... \vee Q_n$ where each Q$_i$ is either P$_i$ or $\neg p_i$

2.6.4 Principal Conjunctive Normal form :

A formula α is in principle conjugate normal form if α is a product of max terms. For obtaining the principle conjunctive normal form of α we can construct the principle disjunctive normal form of $\neg\, \alpha$ and apply negation.

Example 18

Obtain a conjunctive normal form of

1. $\alpha = P \vee (Q \rightarrow R)$
2. $\alpha = (\neg P \rightarrow R) \wedge (P \leftrightarrow Q)$

1) Consider

$\alpha = P \vee (Q \rightarrow R)$

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

$\equiv \neg(P \vee (\neg Q \vee R))$ Implication law

$\equiv \neg P \wedge (\neg(\neg Q \vee R))$ De-Morgans law

$\equiv \neg P \wedge (Q \wedge \neg R)$ De-Morgans law & Double negation

$\therefore \alpha \equiv \neg P \wedge (Q \wedge \neg R)$

Hence, this is the required conjunctive normal form.

The principal conjugate normal form of α is $\neg(\neg p \wedge (Q \wedge \neg R)) = P \vee \neg Q \vee R$

2) $\alpha = (\neg P \rightarrow R) \wedge (P \leftrightarrow Q)$

Since, we know that

$P \leftrightarrow Q \equiv \neg P \vee Q$ Implication law

$P \rightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$ Implication law

$\therefore \alpha \equiv (\neg P \rightarrow R) \wedge ((P \rightarrow Q) \wedge (Q \rightarrow P))$

$$\equiv \left(\neg(\neg P)\vee R\right)\wedge\left(\left(\neg P\vee Q\right)\wedge\left(\neg Q\vee P\right)\right)$$

$$\equiv \left(P\vee R\right)\wedge\left(\left(\neg P\vee Q\right)\wedge\left(\neg Q\vee P\right)\right)$$

$$\alpha \equiv \left(P\vee R\right)\wedge\left(\neg P\vee Q\right)\wedge\left(\neg Q\vee P\right)$$

Which is required conjunctive normal form.

2.7 PREDICATES AND QUANTIFIERS

2.7.1 Predicates : A predicate is a statement containing one or more variables.

Proposition :

If values are assigned to all the variables in a predicate, the resulting statement is a proposition.

e.g.

1. $x < 9$ is a predicate
2. 4 < 9 is a proposition

Propositional Function :

Let a be a given set. A propositional function (or : on open sentence or condition) defined on A is an expression P(*x*) which has the property that P(a) is true or false for each $a \in A$.

The set A is called domain of P(*x*) and the set T_p of all elements of A for which P (a) is true is called the truth set of P(*x*).

i.e. $T_p = \{x : x \in A, p(x)\text{is true}\}$ or T_p = {x : p(*x*)} Another use of predicates is in programming Two common constructions are “if P(x), then execute certain steps” and “while Q(x), do specified actions.” The predicates P(x) and Q(x) are called the guards for the block of programming code often the guard for a block is a conjunction or disjunction.

e.g. Let A = {*x* / *x* is an integer < 8}
Here P(*x*) is the sentence “*x* is an integer less than 8”.

The common property is “an integer less than 8”.

$\therefore$ P(1) is the statement “1 is an integer less than 8”.
$\therefore$ P(1) is true, $\therefore 1 \in A$ etc.

2.7.2 Quantifiers :

The expressions ‘ for all’ and ‘there exists’ are called quantifiers. The process of applying quantifier to a variable is called quantification of variables.

Universal quantification :

The universal quantification of a predicate P(*x*) is the statement, “For all values of *x*, P(*x*) is true.”

The universal quantification of P(*x*) is denoted by $\forall$ for all x P(*x*).

The symbol $\forall$ is called the universal quantifier.

e.g.

1) The sentence P(x) : - (-*x*) = *x* is a predicate that makes sense for real numbers *x*.

The universal quantification of P(*x*), $\forall$ *x* P(*x*) is a true statement because for all real numbers, -(- *x*) = *x*.

2) Let Q(*x*) : x + 1 < 5, then $\forall$ Q(*x*) : x + < 5 is a false statement, as Q(5) is not true. Universal quantification can also be stated in English as “for every *x*”, “every *x*”, or “for any *x*.”

Existential quantification -

The existential quantification of a predicate P(*x*) is the statement “There exists a value of *x* for which P(*x*) is true.”

The existential quantification of P(*x*) is denoted $\exists x P(x)$. The symbol $\exists$ is called the existential quantifier.

e.g.

1) Let $Q : x+1<4$. The existential quantification of Q(*x*), $\exists x Q(x)$ is a true statement, because Q(2) is true statement.

2) The statement $\exists y$, y + 2 = y is false. There is no value of y for which the propositional function y+2=y produces a true statement.

Negation of Quantified statement :

$\neg\left(\forall\, x \in a\right) p(x) \equiv \left(\exists\, x \in A\right) \neg\, p(x)$

or $\neg \forall\, x\, p(x) \equiv \exists\, x \neg\, p(x)$

This is true for any proposition p(*x*). DeMorgan’s Law.

2.7.3 The result for universal and existential quantifiers is as follows.

I) $\neg\left(\forall x \in A\right) p(x) \equiv \left(\exists x \in A\right) \neg p(x)$

In other words, the following two statements are equivalent.

i) It is not true that, for all $a \in A$, P(a) is true.

ii) There exists an $a \in A,$ such that P(a) is false.

II) $\neg\left(\exists x \in A\right) p(x) \equiv \left(\forall x \in A\right) \neg p(x)$

That is, the following two statements are equivalent.

i) It is not true that for some $a \in A,$ P(a) is true.
ii) For all $a \in A,$ P(a) is false.

Other several properties for the universal and existential quantifiers are………

III) $\exists x(p(x) \Rightarrow Q(x) \equiv$ ~~v~~ $xP(x) \Rightarrow \exists x q(x)$

IV) $\exists x(P(x) \wedge Q(x)) \Rightarrow \exists x P(x) \wedge \exists x Q(x)$ is a tautology.

V) $((\forall x p(x)) \vee (\forall x Q(x)) \Rightarrow \forall x(p(x) \vee Q(x))$ is a tautology.

VI) $\forall x(P(x) \wedge Q(x) \equiv \forall x P(x) \wedge \forall x Q(x)$

VII) $\exists x(P(x) \vee Q(x) \equiv \exists x(P(x) \vee \exists x Q(x)$

Example 19 :

Express the statement using quantifiers. “Every student in your school has a computer or has a friend who has a computer.”

Solution :

Let c(*x*) : “*x* has a computer”
F(*x*,*y*) : “*x* and y are friends”

$\therefore$ We have
$\forall x(c(x) \vee \exists y(c(y) \wedge F(x, y))$

Example 20 :

Express following using quantifiers.
i) There exists a polar bear whose colour is not white.
ii) Every polar bear that is found in cold region has a white colour.

Solution :

Let A(*x*) : *x* has a white colour
B(*x*) : *x* is a polar bear.
C(*x*) : *x* is found in cold region.
Over the universe of animals.

i) There exists a polar bear whose colour is not white.
$\exists x(B(x) \rightarrow \neg A(x))$

ii) Every polar bear that is found in cold regions has a white colour.
$\forall x((B(x) \wedge c(x)) \rightarrow A(x))$.

2.8 THEORY OF INFERENCE FOR THE PREDICATE CALCULAS

If an implication $P \Rightarrow Q$ is a tautology where P and Q may be compound statement involving any number of propositional variables we say that Q logically follows from P. Suppose $P(P_1, P_2 ....P_n) \rightarrow Q$. Then this implication is true regardless of the truth values of any of its components. In this case, we say that q logically follows from P_1, P_2…..,P_n.

Proofs in mathematics are valid arguments that establish the truth of mathematical statements.

To deduce new statements from statements we already have, we use rules of inference which are templates for constructing valid arguments. Rules of inference are our basic tools for establishing the truth of statements. The rules of inference for statements involving existential and universal quantifiers play an important role in proofs in Computer Science and Mathematics, although they are often used without being explicitly mentioned.

2.8.1 Valid Argument :

An argument in propositional logic is a sequence of propositions.

All but the final propositions in the argument are called hypothesis or Premises.

The final proposition is called the conclusion.

An argument form in propositional logic is a sequence of compound propositions - involving propositional variables.

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

2.8.2 Rules of Inference for Propositional logic

We can always use a truth table to show that an argument form is valid. Arguments based on tautologies represent universally correct method of reasoning. Their validity depends only on the form of statements involved and not on the truth values of the variables they contain such arguments are called <u>rules of inference</u>.

These rules of inference can be used as building blocks to construct more complicated valid argument forms

e.g.
Let P: "You have a current password"
Q: "You can log onto the network".

Then, the argument involving the propositions,
"If you have a current password, then you can log onto the network".

"You have a current password" therefore: You can log onto the network" has the form …
.

$$\begin{array}{l} P \to Q \\ \underline{\ \ P\ \ } \\ \therefore Q \end{array}$$

Where $\therefore$ is the symbol that denotes 'therefore we know that when P & Q are proposition variables, the statement $((P \to Q) \wedge P) \to Q$ is a tautology.

$\therefore$ This is valid argument and hence is a rule of inference, called modus ponens or the law of detachment.

(Modus ponens is Latin for mode that affirms)
The most important rules of inference for propositional logic are as follows…..

	Rule of Inference	Tautology	Name
1)	P $\underline{P \to Q}$ $\therefore Q$	$(P \wedge (P \to Q)) \to Q$	Modus ponens
2)	$\neg Q$ $\underline{P \to Q}$ $\therefore \neg P$	$[\neg Q \wedge (P \to Q)] \to \neg P$	Modus tollens
3)	$P \to Q$ $\underline{Q \to R}$ $\therefore P \to R$	$[(P \to Q) \wedge (Q \to R)] \to (P \to$	Hypothetical syllogism
4)	$P \vee Q$ $\underline{\neg P}$ $\therefore Q$	$[(P \vee Q) \wedge \neg P] \to Q$	Disjunctive syllogism

5)	$\dfrac{P}{\therefore PVQ}$	$P \to (P \vee Q)$	Addition
6)	$\dfrac{P \wedge Q}{\therefore P}$	$(P \wedge Q) \to P$	Simplification
7)	P $\dfrac{Q}{\therefore P \wedge Q}$	$((P) \wedge (Q)) \to P \wedge Q$	Conjunction
8)	$P \vee Q$ $\dfrac{\neg P \vee R}{\therefore Q \vee R}$	$\left[(P \vee Q) \wedge (\neg P \vee R)\right] \to (Q \vee R$	Resolution

Example 21 :

Show that $R \to S$ can be derived from the premises (i) $P \to (Q \to S)$ (ii) $\neg (R \vee P)$ and iii) Q.

Solution :

The following steps can be used to establish the conclusion.

	Steps	Reason
1	$P \to (Q \to S)$	Premise (i)
2	$R \vee P$	Premise (ii)
3	$R \to P$	Line 2, implication
4	$R \to (Q \to S)$	Hypothetical Syllogism
5	$R \to \left(\neg Q \vee S\right)$	Line 4, implication
6	$\neg R \vee \left(\neg Q \vee S\right)$	Line 5, implication
7	Q	Premise (iii)
8	$\neg R \vee S$	Line 6, 7 and Disjunctive syllogism
9	R$\to S$	Line 8, implication

Hence the proof :

Example 22 :

Test the validity of the following arguments :

1. If milk is black then every crow is white.
2. If every crow is white then it has 4 legs.
3. If every crow has 4 legs then every Buffalo is white and brisk.
4. The milk is black.
5. So, every Buffalo is white.

Solution :

Let P : The milk is black
Q : Every crow is white
R : Every crow has four legs.
S : Every Buffalo is white
T : Every Buffalo is brisk

The given premises are

(i) $P \to Q$

(ii) $Q \to R$

(iii) $R \to S \wedge T$

(iv) P

The conclusion is S. The following steps checks the validity of argument.

1.	$P \to Q$	… premise (1)
2.	$Q \to R$	… Premise (2)
3.	$P \to R$	… line 1. and 2. Hypothetical syllogism (H.S.)
4.	$R \to S \wedge T$	… Premise (iii)
5.	$P \to S \wedge T$	… Line 3. and 4.. H.S.
6.	P	… Premise (iv)
7.	$S \wedge T$	Line 5, 6 modus ponets
8.	S	Line 7, simplification
∴	The argument is valid	

Example 23 :

Consider the following argument and determine whether it is valid or not. Either I will get good marks or I will not graduate. If I did not gradute I will go to USA. I get good marks. Thus, I would not go to USA.

Solution :

Let P : I will get good marks.
Q : I will graduate.
R : I will go to USA

The given premises are

i) $P \vee \neg Q$

ii) $\neg Q \to R$

iii) P

The conclusion is $\neg$ R.
The following steps checks is validity.

	Steps	Reason
1.	$P \vee \neg Q$	… premise (i)
2.	$\neg\neg P \vee \neg Q$	…Double negation
3.	$\neg P \to \neg Q$	Line 2, Implication
4.	$\neg Q \to R$	… premise (ii)

5.	$\neg P \rightarrow R$	Line 3, 4, H.S.
6.	P	Premise (iii)
7.	R	Line 5 implication and line 6
8.	Conclusion is R or $\neg$ R	Line 7 simplification

$\therefore$ The argument is not valid

2.9 MATHEMATICAL INDUCTION

Here we discuss another proof technique. Suppose the statement to be proved can be put in the from $\forall\, n \geq n_0$. P(n), where n_0 is some fixed integer.

That is suppose we wish to show that P(n) is true for all integers $n \geq n_0$.

The following result shows how this can be done.

Suppose that

(a) $P(n_0)$ is true and

(b) If P(K) is true for some $K \geq n_0$, then P(K + 1) must also be true. The P(n) is true for all $n \geq n_0$.

This result is called the principle of Mathematical induction.

Thus to prove the truth of statement $\forall\, n \geq n_0$. P(n), using the principle of mathematical induction, we must begin by proving directly that the first proposition $P(n_0)$ is true. This is called the basis step of the induction and is generally very easy.

Then we must prove that $P(K) \Rightarrow P(K + 1)$ is a tautology for any choice of $K \geq n_0$. Since, the only case where an implication is false is if the antecedent is true and the consequent is false; this step is usually done by showing that if P(K) were true, then P(K + 1) would also have to be true. This step is called induction step.

In short we solve by following steps.

1. Show that P(1) is true.
2. Assume P(k) is true.
3. Prove that P(k +1) is true using P(k)

Hence P(n) is true for every n.

Example 24 :

Using principle of mathematical induction prove that…

1) $1 + 2 + 3 + \ldots.. + n = \dfrac{n(n+1)}{2}$ for all $n \geq 1$

2) $n^3 - n$ is divisible by 3 for $n \in Z^+$

3) $2^n > n$ for all positive integers n.

4) $n! \geq 2^{n-1}$

5) If A_1, A_2, …… A_n be any n sets then $\overline{\left(\bigcup_{i=1}^{n} A_i\right)} = \bigcap_{i=1}^{n} \overline{A_i}$

Solution :

For all n, 1) Let $P(n): 1+2+3+---+n = \dfrac{n\,(n+1)}{2}, n \geq 1$

__Step 1__ : Here $n_0 = 1$

We must show that P (1) is true.

P (1) is the statement

$1 = \dfrac{1(1+1)}{2}$

Which is clearly true.

Hence P(1) is true.

__Step 2__ :
Assume P(K) is true for K ≤ n.

$\therefore$ $P(K) \equiv 1+ 2 + \ldots.. + K = \dfrac{K(K+1)}{2}$ $K \geq 1$ …..(1)

__Step 3__ :
To show that P(K + 1) is true.

$\therefore$ $P(K+1) = 1 + 2 + \ldots.. + (K + 1) = \dfrac{(K+1)((K+1)+1)}{2}$

Consider,

$1 + 2 + \ldots.. + (K + 1) = 1 + 2 + \ldots.. + K + (K +1)$

$= \dfrac{K(K+1)}{2} + (K + 1)$ using eqn. (1)

$\therefore$ $1 + 2 + \ldots.. + (K + 1) = \dfrac{K(K+1)+2(K+1)}{2}$

$= \dfrac{(K+1)+(K+2)}{2}$

$= \dfrac{(K+1)+((K+1)+1)}{2}$

Which is RHS of P(K + 1)
Thus, P(K + 1) is true.

$\therefore$ By principle of mathematical induction it follows that P(n) is true for all n≥1.

$\therefore$ $1 + 2 + \ldots.. + n = \dfrac{n(n+1)}{2}$

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

<u>Step 1</u> : We note that,
P(1) : $1^3 - 1 = 0$ is divisible by 3
$\therefore$ P(1) is true.

<u>Step 2</u> :
Assume P(K) is true for $K \leq n$
$\therefore$ P(K): $K^3 - K$ is divisible by 3.
We can write $K - k = 3m$ for $m \in N$. ……(1)

<u>Step 3</u> :
We prove that P(K + 1) is true.
P(K + 1); $(K + 1)^3 - (K + 1)$ is divisible by 3.
Consider

$$\begin{aligned}(K+1)^3 - (K+1) &= K^3 + 3K^2 + 3K + 1 - K - 1 \\ &= K^3 + 3K^2 + 2K \\ &= 3m + K + 3K^2 + 2K \quad \text{(using (1))} \\ &= 3(m + K + K^2)\end{aligned}$$

Hence $(K + 1)^3 - (K + 1)$ is divisible by 3.

Thus, P(K + 1) is true when P(K) is true.

$\therefore$ By principle of mathematical induction the statement is true for every positive integer n.

3) Let P(n) : $2^n > n$ $\forall$ positive integer n.

<u>Step I</u> : For n = 1, $2^1 = 2 > 1$
Hence P(i) is true.

<u>Step II</u> : Assume P(K) is true for every positive integer K i.e.

$$2^K > K \quad …..(1)$$

<u>Step III</u> : To show that P(K + 1) is true
From (1),

$$2^K > K$$

Multiplying both sides by 2, we get,

$$2.2^K > 2.K$$
$$\therefore \quad 2^{K+1} > 2K$$
$$\therefore \quad 2^{K+1} > K + K > K + 1$$

$\therefore$ P(K + 1) is true when P(K) is true. Hence, by principle of mathematical induction, P(n) is true for every positive integer n.
$\therefore$ $2^n > n$ for positive integer n.

4) Let P(n) : $n! \geq 2^{n-1}$

Step I : For n = 1

$1! = 1 \geq 2^{1-1} = 2^0 = 1$

$\therefore$ P(1) is true.

Step II : Assume P(K) is true for some K < n.

$\therefore\ K! \geq 2^{k-1}$ …..(1)

Step III : Prove that P(K + 1) is true.

Consider $K! \geq 2^{k-1}$ (from (1))

As $K + 1 \geq 2$

$\therefore$ $K! \geq 2^{k-1}$ and $K + 1 \geq 2$

Taking the product we get,

$K! \times (K + 1) \geq 2^{k-1} \times 2$

$\therefore$ $(K + 1)K! \geq 2^{k-1+1}$

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

Hence P(K + 1) is true.

$\therefore$ By principle of mathematical induction P(n) is true for every n.

5) Let $P(n) : \overline{\left(\bigcup_{i=1}^{n} A_i\right)} = \bigcap_{i=1}^{n} \overline{A_i}$

Step I : For n = 2,

LHS $= \overline{\left(\bigcup_{i=1}^{2} A_i\right)} = \overline{(A_1 \cup A_2)} = \overline{A}_1 \cap \overline{A}_2$

& RHS $= \bigcap_{i=1}^{2} \overline{A_i} = \overline{A}_1 \cap \overline{A}_2$

$\therefore$ LHS = RHS

Hence P(2) is true.

Step 2 : Assume P(K) is true for some K < n

$\therefore$ $\overline{\left(\bigcup_{i=1}^{k} A_i\right)} = \bigcap_{i=1}^{k} \overline{A_i}$ …..(1)

Step 3 : Prove that P(K + 1) is true.

Consider

$$\overline{\left(\bigcup_{i=1}^{k+1} A_i\right)} = \overline{\left(\bigcup_{i-1}^{k} A_i \cup A_{k+1}\right)} = \overline{\left(\bigcup_{i-1}^{k} A_i\right)} \cap \overline{A_{k+1}}$$

$$= \bigcap_{i=1}^{k} \overline{A_i} \cap A_{k+1} \text{ (from (1))}$$

$$= \bigcap_{i=1}^{k+1} \overline{A_i}$$

$\therefore$ P(K + 1) is true

∴ By principle of mathematical induction P(n) is true for all n.

∴ $\left(\overline{\bigcup_{i=1}^{n} A_i}\right) = \bigcap_{i=1}^{n} \overline{A_i}$

2.10 UNIT AND EXERCISE :

1. Construct the truth table of

   $(P \to Q) \leftrightarrow (\neg Q \to \neg P)$ [Jan. 11]

2. Construct the truth table of

   $(Q \wedge P) \vee (Q \wedge \neg P)$ [Dec. 09]

3. Construct the truth table for each of the following.

   i) $(P \to Q) \vee (\neg P \to Q)$

   ii) $P \leftrightarrow \neg P$

   iii) $(P \vee Q) \wedge \neg R$

   iv) $P \to (\neg Q \vee R)$

   v) $(PQ) \wedge (\neg P \to R)$

4. Show that P V $P \vee (Q \wedge R)$ and $(P \vee Q) \wedge (P \vee R)$ are logically equivalent.

5. Show that $(\neg P \wedge (\neg Q \wedge R)) \vee (Q \wedge R) \vee (P \wedge R) \equiv R$ [Jan. 11]

6. Show that $(P \wedge Q) \to (P \vee Q)$ is a tautology.

7. Determine whether $(P \to Q) \wedge (Q \to R) \to (P \to R)$ is a tautology or contradiction or neither. [May 10]

8. Obtain the conjunctive normal form of

   $\neg(P \vee Q) \leftrightarrow (P \wedge Q)$ [Jan. 2011]

9. Obtain conjunctive and disjunctive normal form of the following.

   i) $(P \wedge Q) \vee (\neg P \wedge Q \wedge R)$ [May 10]

   ii) $(\neg P \vee \neg Q) \to (P \leftrightarrow Q)$ [Dec. 09]

   iii) $P \vee (Q \to R)$

   iv) $\neg (P \vee Q) \leftrightarrow (P \wedge Q)$

   v) $Q \vee (P \wedge \neg Q) \vee (\neg P \wedge \neg Q)$

10. Objtain principle disjunctive and conjunctive normal form of

    i) $(\neg P \vee \neg Q) \to (P \leftrightarrow \neg Q)$

    ii) $(\neg P \vee \neg Q) \to (P \leftrightarrow Q)$

11. Obtain a conjunctive normal form of $Q \vee (P \wedge \neg Q) \vee \left(\neg P \wedge \neg Q\right)$ show that it is a tautology.

12. What is quantifier ? Explain with suitable examples.

13. Check the validity of following argument “If Anand has completed M.C.A. or M.B.A. Then he is assured a good job. If Anand is assured a good job, he is happy. Anand is not happy. So Anand has not completed M.C.A.”

14. Show that conclusion S follows from the premises (P $\rightarrow$ Q) $\wedge$ (P $\rightarrow$ R), $\neg$ (Q $\wedge$ R) and S V P.

15. Express the following using quantifiers.

    i) Every student in the college has a computer or has a friend who has a computer.

    ii) All rational numbers are real numbers.

    iii) Some rational numbers are not real.

    iv) All men are mortal.

    v) Some women are beautiful.

16. Using Principle of mathematical induction prove that $n^3 + 2n$ is divisible by 3 for every positive integer n.

17. Prove by mathematical induction that $2^n < n!$ for $n \geq 4$.

18. Show by mathematical induction that for all $n \geq 11^2 + 2^2 + 3^2 + --- + n^2 = \frac{n(n+)\,(2n+1)}{6}$

19. Prove by mathematical induction that 3 / (n^3 - n) for every positive inter n.

20. Prove by mathematical induction

    i) $5^n + 3$ is divisible by 4.

    ii) $n^2 + n$ is always even.

    iii) Let P(n) : $1^3 + 2^3 + 3^3 + \ldots.. + n^3 = \frac{n^2\,(n+1)^2 + 4}{4}$

    a) Use P(k) to show P(k+1)

    b) Is P(n) true for all $n \geq 1$

❖❖❖❖

3

SET THEORY AND RELATION

Unit Structure

3.0 OBJECTIVES:

1. Definition and examples of sets.
2. Basic operations and diagrammatic representation of sets.
3. Definition of relations and diagraphs
4. Concept of partition and its relationship with equivalence relation.

3.1 INTRODUCTION:

In the school, we have already studied sets along with the properties of the sets. In this chapter, we revise the concept and further, discuss the concept of an algebraic property called relation.

Set Theory, branch of mathematics concerned with the abstract properties of sets, or collections of objects. A set can be a physical grouping, such as the set of all people present in a room; or a conceptual aggregate, such as the set of all British prime ministers, past and present. Each of these sets is defined by a property that its members share, but it is possible for a set to be a completely arbitrary collection.

Set theory was first given formal treatment by the German mathematician Georg Cantor in the 19th century. The set concept is one of the most basic in mathematics, explicitly or implicitly, in every area of pure and applied mathematics, as well as Computer science.

Relationships between elements of sets occur in many contexts. We deal with many relationships such as student's name and roll no., teacher and her specialisation, a person and a relative (brother – sister, mother – child etc.) In this section, we will discuss mathematical approach to the relation. These have wide applications in Computer science (e.g. relational algebra)

3.2. DEFINITIONS AND REPRESENTATION OF SETS:

Definition 3.2.1: Set is an unordered collection of objects.
The object in a set is called as an element or member.

We denote sets by capital letters such as A, B, C and elements by small letters. Typically sets are described by two methods

i. Roster or list method:
   In this method, all the elements are listed in braces. E.g.
   A = {2, 3, 5, 7, 11, 13 }
   N = { 2, 4, 6, ... }

ii. Set-Builder method:
   In this method, elements are described by the property they satisfy. E.g.
   $A = \{ x : x$ is a prime number less than 15}
   $B = \{ x : x = 2n, n \in$ N }

Definition 3.2.2: A set containing no element is called as an **empty set**. E.g. Set of even prime numbers greater than 10.
Empty set is denoted by { } or ϕ.

Definition 3.2.3: A set *A* is said to be a **subset** of set *B*, if every element of *A* is also an element of *B*. It is denoted by '$\subseteq$ '
$A \subseteq B$. E.g. A = {1, 2, 3, 4 } and B = { 1, 2, 3, 4, 7, 8 } Then A $\subseteq$ B.

Definition 3.2.4: A set *A* is said to be a **superset** of set *B*, if *B* is a subset of *A*. It is denoted by $A \supseteq B$.

Definition 3.2.5: A set is *A* is said to be a **proper subset** of *B*, if *A* is a subset of *B* and there is at least one element in *B*, which is not an element of *A*. Set A explained in Definition 3.2.3, is a proper subset of *B*.

Definition 3.2.6: A set which contains all objects under consideration is called as **Universal** set and is denoted by *U*.

Note: Two sets are said to be equal if and only if they have same elements. E.g. If $A = \{2, 5, 7, 9\}$ and $B = \{5, 2, 7, 9\}$, then A and B are equal.

Now we shall discuss various operations on sets. For this discussion, let U be universal set and let A and B be two subsets of U.

Definition 3.2.7: Set of all elements in A or in B or in both, is defined as **union** of A and B and is denoted by $A \cup B$.

E.g. If $A = \{1, 2, 3, 5, 7\}$ and $B = \{2, 5, 10\ 11\}$, then
$A \cup B = \{1, 2, 3, 5, 7, 10, 11\}$

Definition3.2.8: Set of all elements, that are common in A as well as in B, is defined as **intersection** of A and B and is denoted by $A \cap B$.

E.g. If $A = \{1, 2, 3, 5, 7\}$ and $B = \{2, 5, 10, 11\}$, then
$A \cap B = \{2, 5\}$.

Definition 3.2.9: Set of all elements, that are in A, but not in B, is called as difference between A and B and denoted by
$A - B$. E.g. If $A = \{1, 4,7,8,9\}$ and $B = \{4,9,11,13\}$ then, $A - B = \{1,7, 8\}$.

Definition 3.2.10: The total number of elements in a set is called as **cardinality** of a set. E.g. If $A = \{2, 3, 5, 7, 11, 13\}$ then, Cardinality of A, denoted by $|A|$, is 6. If a set is infinite, then its cardinality is infinity.

Definition 3.2.11: If U is a universal set and A is its subset, then complement of A, denoted by A^C, is all elements of U, that are not in A. E.g. If $U = \{ x : x \in \mathrm{N}, x \leq 15 \}$ and
$A = \{\mathrm{x} : x \in \mathrm{U} \text{ and } 3 \mid x \}$, then $A^C = \{ 1, 2, 4, 5, 7, 8, 10, 11, 13, 14\}$.

Definition 3.2.12: A power set of a set A, denoted by $P(A)$, is set of all subsets of A. E.g. If $A = \{ 1, 2, 3 \}$, then,
$P(A) = \{ \phi, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\},\{1, 2, 3\}\}$.

Note: If number of elements in A is n, then the number of elements in the power set of A is 2^n.

Definition 3.2.13: Let A and B be two sets. The product set of A and B (or Cartesian product of A and B), denoted by
$A \times B$, is set of all ordered pairs from A and B. Thus,
$A \times B = \{(a, b): a \in A, b \in B\}$.
E.g. Let $A = \{ 1, 2, 3 \}$ and $B = \{ 4, 5\}$ then
$A \times B = \{(1, 4), (1, 5), (2, 4), (2, 5), (3, 4), (3, 5)\}$.

3.3 DIAGRAMMATIC REPRESENTATION OF A SET:

British mathematician, John Venn, devised a simple way to represent set theoretic operations diagrammatically. These diagrams are named after him as Venn Diagrams.

Universal set is represented by a rectangle and its subsets using a circle within it.

In the following figures, basic set theoretic operations are represented using Venn diagrams.

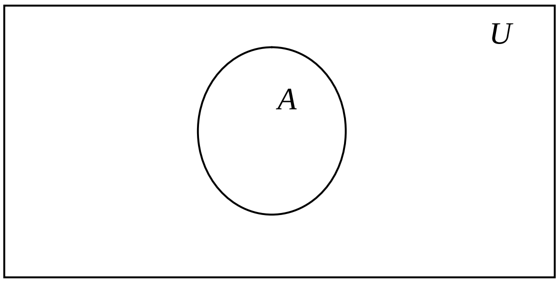


Figure 3.1: *A* is a subset of universal set *U*.

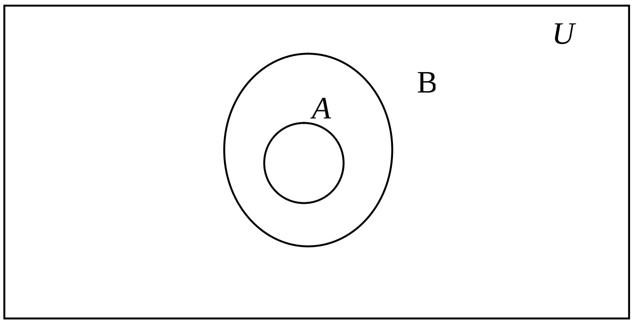


Figure 3.2: $A \subseteq B$

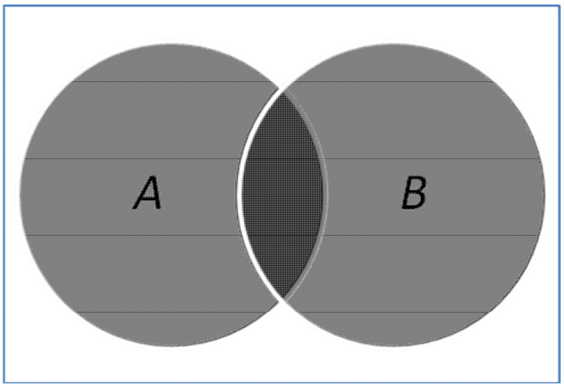


Figure 3.3: $A \cup B$ **: Entire shaded region**
$A \cap B$ **: Dark gray shaded region**

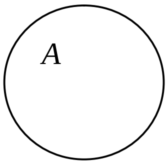


Figure 3.4: A^C, the shaded region

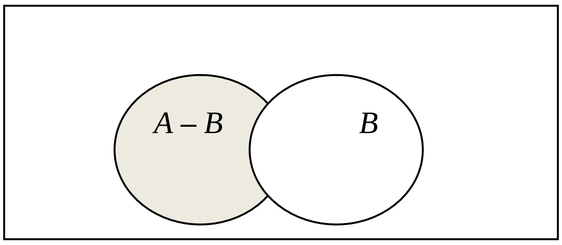


Figure 3.5: *A – B*, the shaded region

3.4 THE ALGEBRA OF SETS:

The following statements are basic consequences of the above definitions, with *A*, *B*, *C*, ... representing subsets of a universal set *U*.

1. $A \cup B = B \cup A$. (Union is commutative)
2. $A \cap B = B \cap A$. (Intersection is commutative)
3. $(A \cup B) \cup C = A \cup (B \cup C)$. (Union is associative)
4. $(A \cap B) \cap C = A \cap (B \cap C)$. (Intersection is associative)
5. $A \cup \phi = A$.
6. $A \cap \phi = \phi$.
7. $A \cup U = U$.
8. $A \cap U = A$.
9. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.(Union distributes over intersection)
10. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$. (Intersection distributes over union)
11. $A \cup A^C = U$.
12. $A \cap A^C = \phi$.
13. $(A \cup B)^C = A^C \cap B^C$. (de' Morgan's law)
14. $(A \cap B)^C = A^C \cup B^C$. (de' Morgan's law)
15. $A \cup A = A \cap A = A$.
16. $(A^C)^C = A$.
17. $A - B = A \cap B^C$.
18. $(A - B) - C = A - (B \cup C)$.
19. If $A \cap B = \phi$, then $(A \cup B) - B = A$.
20. $A - (B \cup C) = (A - B) \cap (A - C)$.

This algebra of sets is an example of a Boolean algebra, named after the 19th-century British mathematician George Boole, who applied the algebra to logic. The subject later found applications in electronics.

3.5 THE COMPUTER REPRESENTATION OF SETS:

There are various ways to represent sets using a computer. Modern programming languages, such as JAVA, have predefined collection class to represent the set. In such class, we need to insert the set elements and there are various class operations defined for the algebraic operations on the set.

In this section, we shall present a method for storing elements using the arbitrary ordering of the elements of a universal set.

Assume that the universal set U is finite (and of reasonable size so that the number of elements in U are not larger than the memory size). First, specify the arbitrary ordering of elements of U, such as a_1, a_2, ..., ..., a_n. Represent a subset A of U with the bit string of length n, where the i^{th} bit in this string is 1 if a_i belongs to A and is 0 otherwise.

E.g. Let $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and A be subset of U containing elements that are multiples of 3 or 5. Thus,

$A = \{3, 5, 6, 9, 10\}$. We shall represent elements of U as per the order given in the above set. Then, A represents a bit string 0010110011.

With this, we have completed basic discussion on set theory and now is the time to check the understanding for the same.

3.6 RELATIONS:

Relationship between elements of sets is represented using a mathematical structure called relation. The most intuitive way to describe the relationship is to represent in the form of ordered pair. In this section, we study the basic terminology and diagrammatic representation of relation.

Definition 3.6.1: Let A and B be two sets. A **binary relation** from A to B is a subset of $A \times B$.

Note 3.6.1: If A, B and C are three sets, then a subset of $A{\times}B{\times}C$ is known as ternary relation. Continuing this way a subset of $A_1{\times}A_2{\times}...{\times}A_n$ is known as n – ary relation.

Note3.6.2: Unless or otherwise specified in this chapter a relation is a binary relation.

Let A and B be two sets. Suppose R is a relation from A to B (i.e. R is a subset of $A \times B$). Then, R is a set of ordered pairs where each first element comes from A and each second element from B. Thus, we denote it with an ordered pair (a, b), where $a \in A$ and $b \in B$. We also denote the relationship with $a\ R\ b$, which is read as a related to b. The ***domain*** of R is the set of all first elements in the ordered pair and the ***range*** of R is the set of all second elements in the ordered pair.

Example 3.1: Let $A = \{ 1, 2, 3, 4 \}$ and $B = \{ x, y, z \}$. Let $R = \{(1, x), (2, x), (3, y), (3, z)\}$. Then R is a relation from A to B.

Example 3.2: Suppose we say that two countries are adjacent if they have some part of their boundaries common. Then, "is adjacent to", is a relation R on the countries on the earth. Thus, we have, (India, Nepal) $\in R$, but (Japan, Sri Lanka) $\notin R$.

Example 3.3: A familiar relation on the set **Z** of integers is "m divides n". Thus, we have, $(6, 30) \in R$, but $(5, 18) \notin R$.

Example 3.4: Let A be any set. Then $A \times A$ and ϕ are subsets of $A \times A$ and hence they are relations from A to A. These are known as universal relation and empty relation, respectively.

Note 3.6.3: As relation is a set, it follows all the algebraic operations on relations that we have discussed earlier.

Definition 3.6.2: Let R be any relation from a set A to set B. The **inverse** of R, denoted by R^{-1}, is the relation from B to A which consists of those ordered pairs, when reversed, belong to R. That is:
$R^{-1} = \{(b, a) : (a, b) \in R\}$

Example 3.5: Inverse relation of the relation in example 1.1 is, $R^{-1} = \{(x, 1), (x, 2), (y, 3), (z, 3)\}$.

3.7 REPRESENTATION OF RELATIONS:

Matrices and graphs are two very good tools to represent various algebraic structures. Matrices can be easily used to represent relation in any programming language in computer. Here we discuss the representation of relation on finite sets using these tools.

Consider the relation in Example 3.1.

$$\begin{array}{c|ccc} & x & y & z \\ \hline 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 1 \\ 4 & 0 & 0 & 0 \end{array}$$

Fig. 3.1

Thus, if *a R b*, then we enter 1 in the cell (*a*, *b*) and 0 otherwise.
Same relation can be represented pictorially as well, as follows:

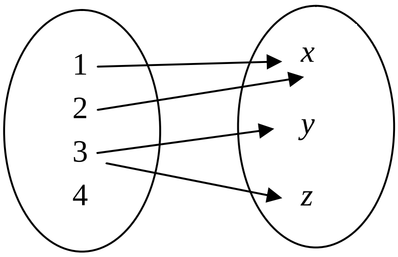


Fig 3.2

Thus, two ovals represent sets *A* and *B* respectively and we draw an arrow from *a* ∈ *A* to *b* ∈ *B*, if *a R b*.

If the relation is from a finite set to itself, there is another way of pictorial representation, known as **diagraph**.

For example, let A = {1, 2, 3, 4} and R be a relation from A to itself, defined as follows:
R = {(1, 2), (2, 2), (2, 4), (3, 2), (3, 4), (4, 1), (4, 3)}
Then, the diagraph of R is drawn as follows:

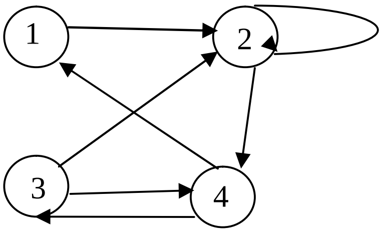


Fig 3.3

The directed graphs are very important data structures that have applications in Computer Science (in the area of networking).

Definition 3.7.1: Let A, B and C be three sets. Let R be a relation from A to B and S be a relation from B to C. Then, composite relation R°S, is a relation from A to C, defined by,
a(R°S)*c*, if there is some b ∈ B, such that *a R b* and *b bsc*.

Example 3.6: Let $A = \{1, 2, 3, 4\}$, $B = \{a, b, c, d\}$, $C = \{x, y, z\}$ and let $R = \{(1, a), (2, d), (3, a), (3, b), (3, d)\}$ and $S = \{(b, x), (b, z), (c, y), (d, z)\}$.

Pictorial representation of the relation in Example 3.6 can be shown as below (Fig 1.4).

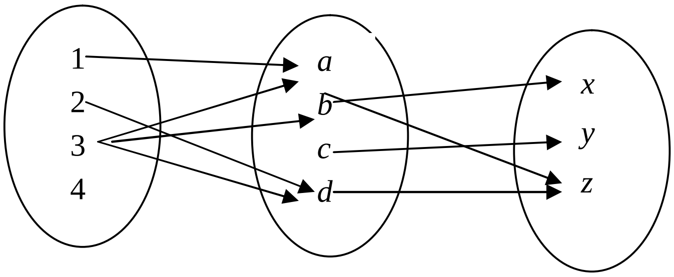


Fig 3.4

Thus, from the definition of composite relation and also from Fig 3.4, $R°S$ will be given as below.

$R°S = \{(2, z), (3, x), (3, z)\}$.

There is another way of finding composite relation, which is using matrices.

Example 3.7: Consider relations R and S in Example 3.6. Their matrix representations are as follows.

$$M_R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad M_S = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Consider the product of matrices M_R and M_S as follows:

$$\mathrm{M_R M_S} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Observe that the non-zero entries in the product tell us which elements are related in $R°S$. Hence, $M_R M_S$ and $M_{R°S}$ have same non-zero entries.

3.8 TYPES OF RELATIONS:

In this section, we discuss a number of important types of relations defined from a set A to itself.

Definition 3.8.1: Let *R* be a relation from a set *A* to itself. *R* is said to be ***reflexive***, if for every $a \in A$, $a\ R\ a$ (*a* is related to itself).

Example 3.8: Let *A* = {a, b, c, d} and *R* be defined as follows:
R = {(*a*, *a*), (*a*, *c*), (*b*, *a*), (*b*, *b*), (*c*, *c*), (*d*, *c*), (*d*, *d*)}.
R is a reflexive relation.

Example 3.9: Let *A* be a set of positive integers and *R* be a relation on it defined as, *a R b* if "*a* divides *b*". Then, *R* is a reflexive relation, as *a* divides to itself for every positive integer *a*.

Note 3.8.1: If we draw a diagraph of a reflexive relation,
then all the vertices will have a loop. Also if we represent reflexive relation using a matrix, then all its diagonal entries will be 1.

Definition 3.8.2: Let *R* be a relation from a set *A* to itself. *R* is said to be ***irreflexive***, if for every $a \in A$, $a\ R\ a$ (*a* is not relat̸ed to itself).

Example 3.10: Let *A* be a set of positive integers and *R* be a relation on it defined as, *a R b* if "*a* is less than *b*". Then, *R* is an irreflexive relation, as *a* is not less than itself for any positive integer *a*.

Example 3.11: Let *A* = {a, b, c, d} and *R* be defined as follows:
R = {(*a*, *a*), (*a*, *c*), (*b*, *a*), (*b*, *d*), (*c*, *c*), (*d*, *c*), (*d*, *d*)}.
Here *R* is neither reflexive nor irreflexive relation as b is not related to itself and *a*, *c*, *d* are related to themselves.

Note 3.8.2: If we draw a diagraph of an irreflexive relation,
then no vertex will have a loop. Also if we represent irreflexive relation using a matrix, then all its diagonal entries will be 0.

Definition 3.8.3: Let *R* be a relation from a set *A* to itself. *R* is said to be ***symmetric***, if for $a, b \in A$, if *a R b* then *b R a*.

Definition 3.8.4: Let *R* be a relation from a set *A* to itself. *R* is said to be ***anti-symmetric***, if for $a, b \in A$, if *a R b* and *b R a*, then *a* = *b*. Thus, *R* is not anti-symmetric if there exists $a, b \in A$ such that *a R b* and *b R a* but $a \neq b$.

Example 3.13: Let *A* = {1, 2, 3, 4} and R be defined as:
R = {(1, 2), (2, 3), (2, 1), (3, 2), (3, 3)}, then R is symmetric relation.

Example 3.14: An equality (or "is equal to") is a symmetric relation on the set of integers.

Example 3.15: Let *A* = {*a*, *b*, *c*, *d*} and *R* be defined as:
R = {(*a*, *b*), (*b*, *a*), (*a*, *c*), (*c*, *d*), (*d*, *b*)}. *R* is not symmetric, as *a R c* but $c \not R a$. *R* is not anti-symmetric, because *a R b* and
b R a, but *a* ≠ *b*.

Example 3.16: The relation "less than or equal to (≤)", is an anti-symmetric relation.

Definition 3.8.5: Let R be a relation defined from a set A to itself. For a, b ∈ A, if a R b, then $b \not R a$, then R is said to be ***asymmetric*** relation.

Example 3.17: Let A = {a, b, c, d} and R be defined as:
R = {(a, b), (b, c), (b, d), (c, d), (d, a)}. R here is asymmetric relation. Because $a \not R b$ but $b \not R a$, $b \not R c$ but $c \not R b$ and so on.

Example 3.18: Relation "is less than (<)", defined on the set of all real numbers, is an asymmetric relation.

Definition 3.8.6: Let *R* be a relation defined from a set *A* to itself. *R* is said to ***transitive***, if for *a*, *b*, *c* ∈ *A*, *a R b* and *b R c*, then *a R c*.

Example 3.19: Let *A* = {*a*, *b*, *c*, *d*} and *R* be defined as follows: *R* = {(*a*, *b*), (*a*, *c*), (*b*, *d*), (*a*, *d*), (*b*, *c*), (*d*, *c*)}. Here *R* is transitive relation on *A*.

Example 3.20: Relation "*a* divides *b*", on the set of integers, is a transitive relation.

Definition 3.8.7: Let *R* be a relation defined from a set *A* to itself. If *R* is reflexive, symmetric and transitive, then *R* is called as **equivalence** relation.

Example 3.21: Consider the set *L* of lines in the Euclidean plane. Two lines in the plane are said to be related, if they are parallel to each other. This relation is an equivalence relation.

Example 3.22: Let *m* be a fixed positive integer. Two integers, *a*, *b* are said to be congruent modulo *m*, written as: *a* ≡ *b* (mod *m*), if *m* divides *a* – *b*. The congruence relation is an equivalence relation.

Example 3.23 : Let $A=\{2,3,4,5\}$ and let $R=\{(2,3),(3,3),(4,5),(5,1)\}$.
Is R symmetric, asymmetric or antisymmetric?
Solution :

a) R is not symmetric, since $(2,3)\in R$, but $(3,2)\notin R$,

b) R is not asymmetric since $(3,3)\in R$

c) R is antisymmetric since if $a\neq b$ either

$(a,b) \notin R$	or	$(b,a) \notin R$
$2 \neq 3$	,	$(3,2) \notin R$
$3 \neq 4$		$(3,4) \notin R$
$4 \neq 5$		$(5,4) \notin R$
$5 \neq 2$		$(2,5) \notin R$

Example 3.24 : Determine whether the relation R on a set A is reflenive, irreflenire, symmetric, asymmetric antisymmetric or transitive.

I) A = set of all positive integers, a R b iff $|a-b| \leq 2$.

[Dec - 02, Nov.-06, May - 07]

Solution :

1) R is reflexive because $|a-a| = 0 < 2, \forall\, a \in A$

2) R is not irreflexive because $|1-1| = 0 < 2$ for $1 \in A$ ($\therefore$ A is the set of all positive integers.)

3) R is symmetric because $|a-b| \leq 2 \Rightarrow |b-a| \leq 2 \ \therefore a\,R\,b \Rightarrow b\,R\,a$

4) R is not asymmetric because $|5-4| \leq 2$ and we have $|4-5| \leq 2$
$\therefore 5\,R\,4 \Rightarrow 4\,R\,5$

5) R is not antisymmetric because $1\,R\,2$ & $2\,R\,1$ $1\,R\,2 \Rightarrow |1-2| \leq 2$ & $2\,R\,1 \Rightarrow |2-1| \leq 2$. But $1 \neq 2$

6) R is not transitive because 5 R 4, 4 R 2 but 5 $\not R$ 2

II) $A = Z^+, a\,R\,b$ iff $|a-b| = 2$ [May - 05]

Solution :

As per above example we can prove that R is not reflexive, R is irrflexive, symmetric, not asymmetric, not antisymmetric & not transitive

III) Let A = {1, 2, 3, 4} and R {(1,1), (2,2), (3,3)} [Dec. - 04]

1) R is not reflexive because $(4,4) \notin R$

2) R is not irreflexive because $(1,1) \notin R$

3) R is symmetric because whenever a R b then b R a.

4) R is not asymmetric because $|R| \Rightarrow |R|$

5) R is antisymmetric because $2\,R\,2, 2\,R\,2 \Rightarrow 2 = 2$

6) R is transitive.

IV) Let $A = Z^+, a\,R\,b$ iff GCD (a, b) = 1 we can say that a and b are relatively prime. [Apr. 04, Nov. 05]

1) R is not reflexive because $(3,3) \neq 1$ it is 3. $\therefore (3,3) \notin R$

2) R is not irreflexive because (1, 1) = 1

3) R is symmetric because for $(a,b)=1 \Rightarrow (b,a)=1$. $\therefore a\,R\,b \rightarrow b\,R\,a$

4) R is not asymmetric because (a, b) = 1 then (b, a) = 1. $\therefore a\,R\,b \rightarrow b\,R\,a$

5) R is not antisymmetric because 2 R 3 and 3 R 2 but $2 \neq 3$.

6) R is not transitive because 4 R 3, 3 R 2 but 4 $\not R$ 2 because (4,2) = G.C.D. (4,2) = $2 \neq 1$.

V) A = Z a R b iff $a \le b+1$ [May 03, May 06]

1) R is reflexive because $a \le a+1 \forall a \in | A$.

2) R is not irreflexive because $0 \le 0+1$ for $\mathrm{O} \in \mathrm{A}$.

3) R is not symmetric because for $2 \le 5+1$ does not imply $5 \le 2+1$.

4) R is not asymmetric because for (2,3) $\in$ R and also (3,2) $\in$ R.

5) R is not antisymmetric because 5 R 4 and 4 R 5 but $4 \neq 5$.

6) R is not transitive because (6,45) $\in$ R, (5,4) $\in$ R but (6,47) $\notin$ R.

3.9 RELATIONS AND PARTITION:

In this section, we shall know what partitions are and its relationship with equivalence relations.

Definition 3.8.1: A partition or a quotient set of a non-empty set *A* is a collection $\mathbb{P}$ of non-empty sets of *A*, such that

(i) Each element of *A* belongs to one of the sets in $\mathbb{P}$.

(ii) If A_1 and A_2 are distinct elements of $\mathbb{P}$, then

$A_1 \cap A_2 = \phi$.

The sets in $\mathbb{P}$ are called the blocks or cells of the partition.

Example 3.23: Let *A* = {1, 2, 3, 4, 5}. The following sets form a partition of *A*, as $A = A_1 \cup A_2 \cup A_3$ and
$A_1 \cap A_2 = \phi$, $A_1 \cap A_3 = \phi$, and $A_2 \cap A_3 = \phi$.
$A_1 = \{1, 2\}$; $A_2 = \{3, 5\}$; $A_3 = \{4\}$.

Example 3.24: Let *A* = {1, 2, 3, 4, 5, 6}. The following sets do not form a partition of *A*, as $A = A_1 \cup A_2 \cup A_3$ but
$A_2 \cap A_3 \neq \phi$.
$A_1 = \{1, 2\}$; $A_2 = \{3, 5\}$; $A_3 = \{4, 5, 6\}$.

The following result shows that if $\mathbb{P}$ is a partition of a set A, then $\mathbb{P}$ can be used to construct an equivalence relation on A.

Theorem: Let $\mathbb{P}$ be a partition of a set *A*. Define a relation *R* on *A* as *a R b* if and only if *a*, *b* belong to the same block of $\mathbb{P}$ then *R* is an equivalence relation on *A*.

Example 3.25: Consider the partition defined in Example 3.23. Then the equivalence relation as defined from the partition is:
R={(1, 1),(1, 2),(2, 1),(2, 2),(3, 3),(3, 5), (5, 3), (5, 5), (4, 4)}.

Now, we shall define equivalence classes of R on a set A.

Theorem: Let R be an equivalence relation on a set A and let a, $b \in A$, then $a\ R\ b$ if and only if $R(a) = R(b)$, where $R(a)$ is defined as: $R(a) = \{x \in A\text{: } a\ R\ x\}$. $R(a)$ is called as **relative set** of a.

Example 3.26: If we consider an example in 3.25, we observe that, $R(1) = R(2)$, $R(3) = R(5)$.

Because R (1) = {1,2}, R (2) = {1,2}, R (3) = {3,5}, R(5) = {3,5}.

Earlier, we have seen that, a partition defines an equivalence relation. Now, we shall see that, an equivalence relation defines a partition.

Theorem: Let R be an equivalence relation on A and let $\mathcal{P}$ be the collection of all distinct relative sets R(a) for a $\in$ A. Then $\mathcal{P}$ is a partition of A and R is equivalence relation of this partition.

Note: If R is an equivalence relation on A, then sets $R(a)$ are called as equivalence classes of R.

Example 3.27: Let $A = \{1, 2, 3, 4\}$ and $R = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 4), (4, 3), (3, 3), (4, 4)\}$. We observe that $R(1) = R(2)$ and $R(3) = R(4)$ and hence $\mathcal{P}$ = { {1, 2}, {3, 4} }.

Example 3.28: Let A = Z (set of integers) and define R as

$R = \{(a, b) \in A \times A\text{: } a \equiv b \pmod 5\}$. Then, we have,

$R(1)$ = {......,–14, –9, –4, 1, 6, 11, ... }

$R(2)$ = {......,–13, –8, –3, 2, 7, 12, ... }

$R(3)$ = {......,–12, –7, –2, 3, 8, 13, ... }

$R(4)$ = {......,–11, –6, –1, 4, 9, 14, ... }

$R(5)$ = {......,–10, –5, 0, 5, 10, 15, ... }.

$R(1)$, $R(2)$, $R(3)$, $R(4)$ and $R(5)$ form partition on Z with respect to given equivalence relation.

3.10 UNIT END EXERCISE:

1. Show that we can have $A \cap B = A \cap C$, without $B = C$.

2. Prove that $(A \cup B) \setminus (A \cap B) = (A \setminus B) \cup (B \setminus A)$. (Note that, this can be used as a definition of $A \oplus B$)

3. Determine whether or not each of the following is a partition of the set N of natural numbers.
   a.[$\{n : n > 5\}$, $\{n : n < 5\}$]
   b.[$\{n : n > 5\}$, $\{ 0 \}$, $\{1, 2, 3, 4, 5\}$]
   c.[$\{n : n^2 > 11\}$, $\{n : n^2 < 11\}$]

4. Suppose $N = \{1, 2, 3, ..., \}$ is a universal set and
   $A = \{x : x \leq 6\}$, $B = \{ x : 4 \leq x \leq 6 \}$,
   $C = \{1, 3, 5, 7, 9\}$, $D = \{2, 3, 5, 7, 8\}$
   Find (i) $A \oplus B$ (ii) $B \oplus C$ (iii) $A \cap (B \oplus D)$
   (iv) $(A \cap B) \oplus (A \cap D)$

5. Let $A = \{1, 2, 3, 4, 6 \}$ and R be the relation on A defined by "x divides y", written an $x \mid y$.
   a. Write R as a set of ordered pairs.
   b. Draw a directed graph of R.
   c. Write down the matrix of relation R.
   d. Find the inverse relation R^{-1} of R and describe it in words.

6. Give an example of relations $A = \{1, 2, 3\}$ having the stated property.
   a. R is both symmetric and antisymmetric
   b. R is neither symmetric nor antisymmetric
   c. R is transitive but $R \cup R^{-1}$ is not transitive.

7. Let A be a set of non-zero integers and let = be the relation on $A \times A$ defined by $(a, b) = (c, d)$, whenever $ad = bc$. Prove that = is an equivalence relation.

8. Prove that if R is an equivalence relation on a set A, then R^{-1} is also an equivalence relation on A.

❖❖❖❖

4

PARTIAL ORDER RELATION

Unit Structure

4.0 OBJECTIVES:

- Definition and examples of partial order relation.
- Representation of posets using Hasse diagram.
- Definition of a Lattice.

4.1 INTRODUCTION:

We often use relation to describe certain ordering on the sets. For example, lexicographical ordering is used for dictionary as well as phone directory. We schedule certain jobs as per certain ordering, such as priority. Ordering of numbers may be in the increasing order.

In the previous chapter, we have discussed various properties (reflexive etc) of relation. In this chapter we use these to define ordering of the sets.

Definition 4.1.1: A relation R on the set A is said to be ***partial order relation***, if it is reflexive, anti-symmetric and transitive.

Before we proceed further, we shall have a look at a few examples of partial order relations.

Example 4.1: Let $A = \{a, b, c, d, e\}$. Relation R, represented using following matrix is a partial order relation.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Observe the reflexive, anti-symmetric and transitive properties of the relation from the matrix.

Example 4.2: Let *A* be a set of natural numbers and relation *R* be "less than or equal to relation (≤)". Then *R* is a partial order relation on *A*. For any $m, n, k \in N$, $n \le n$ (reflexive); if $m \le n$ and $m \ge n$, then $m = n$ (anti-symmetric); lastly, if $m \le n$ and $n \le k$, then $m \le k$ (transitive).

Definition 4.1.2: If *R* is a partial order relation on a set *A*, then *A* is called as partial order set and it is denoted with (*A*, *R*). Typically this set is termed as ***poset*** and the pair is denoted with (*A*, ≤).

4.2 DIAGRAMMATIC REPRESENTATION OF PARTIAL ORDER RELATIONS AND POSETS:

In the previous chapter, we have seen the diagraph of a relation. In this section, we use the diagraphs of the partial order relations, to represent the relations in a very suitable way known as Hasse diagram.

We understand the Hasse diagrame, using following example.

Example 4.3: Let $A = \{a, b, c, d, e\}$ and the following diagram represents the diagraph of the partial order relation on *A*.

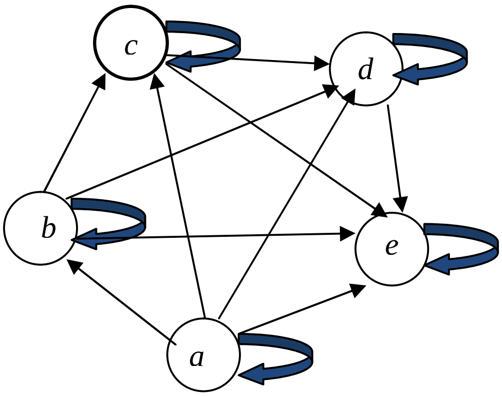


Fig. 4.1

Now, we shall draw Hasse diagram from the above diagrams using following rules.

(i) Drop the reflexive loops

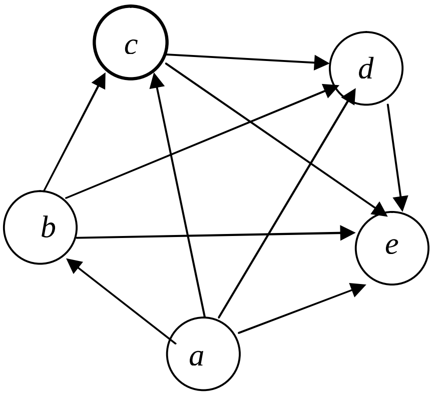


Fig. 4.2

(ii) Drop transitive lines

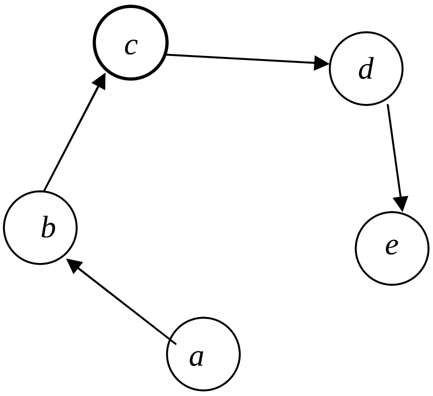


Fig. 4.3

(iii)Drop arrows

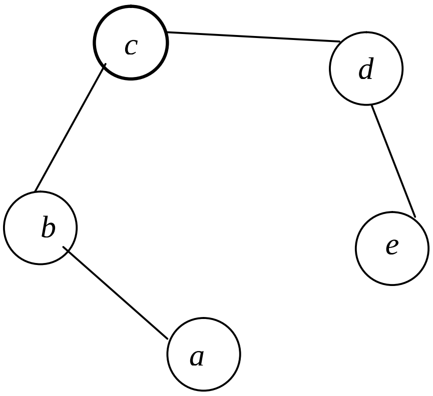


Fig. 4.4

Note 4.1: In many cases, when the graphical representation is so oriented that all the arrow heads point in one direction (upward, downward, left to right or right to left). A graphical representation
in which all the arrowheads point upwards, is known as Hasse diagram.

Example 4.4: `Let $A = \{1, 2, 3, 4, 6, 9\}$ and relation R defined on A be "*a* divides *b*". Hasse diagram for this relation is as follows:

Note 4.2: The reader is advised to verify that this relation is indeed a partial order relation. Further,
arrive at the following Hasse diagram from the diagraph of a relation as per the rules defined earlier.

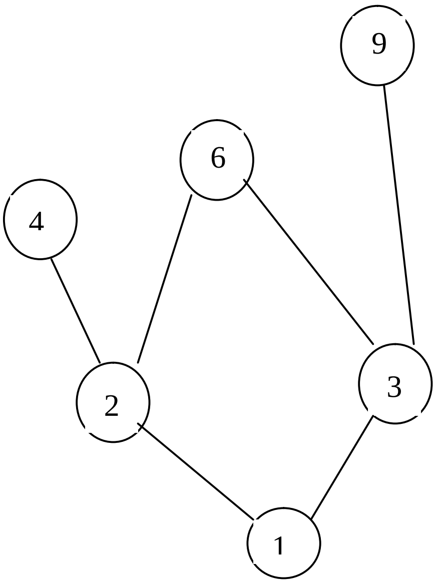


Fig.4.5

Example 4.5 : Determine the Hasse diagram of the relation on A = {1,2,3,4,5} whose M_R is given below :

$$M_R = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution :

Reflexivity is represented by 1 at diagonal place. So after removing reflexivity R is R = {(1,3), (1,4), (1,5), (2,3), (2,4), (2,5), (3,4), (3,5)}

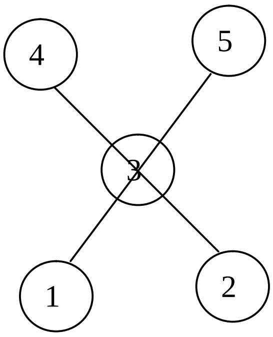
Remove transitivity as

$(1,3)(3,4) \in R$ $\therefore$ remove $(1,4) \in R$

$(2,3)(3,5) \in R$ $\therefore$ remove $(2,5) \in R$ and so on.

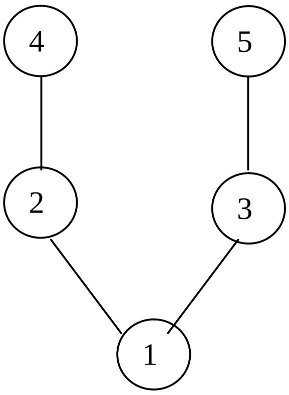
$\therefore R = \{(1,3),(2,3),(3,4),(3,5)\}$

The Hasse Diagram is



Example 4.6 :

Determine matrix of partial order whose Hasse diagram is given as follow -



Solution :

Here A = [1, 2, 3, 4, 5)

Write all ordered pairs (a, a) $\forall\, a \in A$ i.e. relation is reflexive.

Then write all ordered pairs in upward direction. As (1, 2) $\in R$ & (2,4) $\in R \Rightarrow (1,4) \in R$ since R is transitive.

$\therefore R = \{(1,1),(2,2),(3,3),(4,4),(5,5),(1,2),(2,4),(2,4),(1,4),(1,3),(3,5),(1,5)\}$

The matrix M_R can be written as -

$$M_R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Now, we shall have a look at certain terms with reference to posets.

Definition 4.2.1: Let (*A*, ≤) be a partially ordered set. Elements *a*, *b* ∈ *A*, are said to be comparable, if $a \le b$ or $b \le a$.
E.g. In example 4.4, 2 and 4 are comparable, whereas 4 and 9 are not comparable.

Definition 4.2.2: Let (*A*, ≤) be a partially ordered set. A subset of *A* is said to be a ***chain*** if every two elements in the subset are related.

Example 4.7: In the poset of example 4.4, subsets {1, 2, 4}; {1, 3, 6}; {1, 2, 6} and {1, 3, 9} are chains.

Definition 4.2.3: A subset of a poset A is said to be ***anti-chain***, if no two elements of it are related.

Example 4.8: In the poset of example 4.4, subsets {2, 9}; {3, 4}; {4, 6, 9} are anti-chains.

Definition 4.2.4: A partially ordered set *A* is said to be ***totally ordered*** if it is chain.

Example 4.9: Let *A* = {2, 3, 5, 7, 11, 13, 17, 19} and the relation defined on A be ≤. Then poset (*A*, ≤) is a chain.

4.3 MAXIMAL, MINIMAL ELEMENTS AND LATTICES:

In this section, we discuss certain element types in the poset and hence a special kind of poset, Lattice.

To understand these types, we shall refer to the following figures, i.e. Fig.4.6 and Fig.4.7.

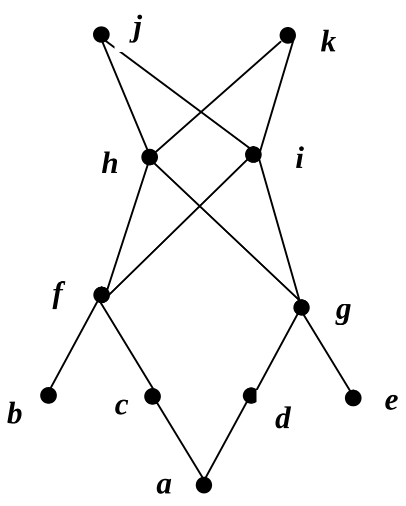


Fig. 4.6

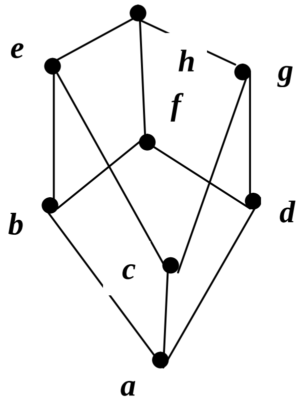


Fig. 4.7

Definition 4.3.1: Let (A, ≤) be a poset. An element $a \in A$ is called a ***maximal element***, if for no $b \in A$, $a \neq b$, $a \leq b$. E.g. In Fig. 4.6, j and k are maximal elements.

Definition 4.3.2: Let (A, ≤) be a poset. An element $a \in A$ is called a ***minimal element***, if for no $b \in A$, $a \neq b$, $b \leq a$. E.g. In Fig. 4.6, a, b and e are minimal elements.

Definition 4.3.3: Let a, b be two elements in the poset (A, ≤). An element $c \in A$, is said to be an ***upper bound*** of a, b if $a \leq c$ and $b \leq c$. E.g. In Fig 4.7, f_1 h are upper bounds of b and d.

Definition 4.3.4: Let a, b be two elements in the poset (A, ≤). An element $c \in A$, is said to be a ***least upper bound*** of a, b if $a \leq c$ and $b \leq c$ and if d is an upper bound of a, b, then $c \leq d$. E.g. In Fig 2.7, f is a least upper bound of b and d.

Definition 4.3.5: Let a, b be two elements in the poset (A, ≤). An element $c \in A$, is said to be a ***lower bound*** of a, b if $c \leq a$ and $c \leq b$. E.g. In Fig 4.6, f, g are lower bounds of h and i.

Definition 4.3.6: Let a, b be two elements in the poset (A, ≤). An element $c \in A$, is said to be a ***greatest lower bound*** of a, b if $c \leq a$ and $c \leq b$ and if d is a lower bound of a, b, then $d \leq c$. E.g. In Fig 4.7, c is a greatest lower bound of e and g.

Definition 4.3.7: A poset (A, ≤) is said to be a ***lattice***, if every two elements in A have a unique least upper bound and a unique greatest lower bound.

E.g. Fig. 4.6 is not a lattice, because j and k are two least upper bounds of h and i, whereas Fig. 4.7 is a lattice.

4.4 SOLVED PROBLEMS:

Problem 4.1: Let (S, R) be a poset. Show that (S, R^{-1}) is also a poset. (S, R^{-1}) is called as dual poset of (S, R).

Solution:

(i) Since $a\ R\ a$ (partial order relation is reflexive), $a\ R^{-1}\ a$. (R^{-1} is reflexive).

(ii) Let $a, b \in$ S, where $a \neq b$. If $a\ R\ b$, $b\ R^{-1} a$.

$a\ R\ b \Rightarrow b \not{R}\ a$ (partial order relation is anti-symmetric) $\Rightarrow$ $a \not{R}^{-1}\ b$.

Thus, $b\ R^{-1} a \Rightarrow a\ \ R^{-1}\ b$ (R^{-1} is anti-symmetric).

(iii)If $a\ R\ b \Rightarrow b\ R^{-1}\ a$ and $b\ R\ c \Rightarrow c\ R^{-1}\ b$; by transitivity of partial order relation, we have $a\ R\ c$. Hence $c\ R^{-1}\ a$.

Thus, $c\ R^{-1} b$ and $b\ R^{-1} a \Rightarrow c\ R^{-1} a$. ($R^{-1}$ is transitive).

From (i), (ii) and (iii), R^{-1} is a partial order relation.

Problem 4.2: Find dual of the following posets.

(i) ({0, 1, 2}, R), where R = {(0, 0), (1, 1), (2, 2), (1, 0), (2, 1), (2, 0)}.

R^{-1} = {(0, 0), (1, 1), (2, 2), (0, 1), (1, 2), (0, 2)}.

(ii) (Z, $\geq$) (That is greater than or equal to relation on the set of integers).

Dual is (Z, $\leq$).

(iii)(Z, |). (That is divisibility relation on the set of integers, i.e. a divides b)

Dual is (Z, is divisible by).

Problem 4.3: Which of the following pairs are comparable in the poset (Z^+, |).

(a) 5, 15 (b) 6, 9 (c) 8, 16 (d) 7, 7

Solution: All except for (b).

Problem 4.4: Find two incomparable elements in the posets

(a) (P({0, 1, 2}), $\subseteq$) (where P({0, 1, 2}) is poset of {0, 1, 2})

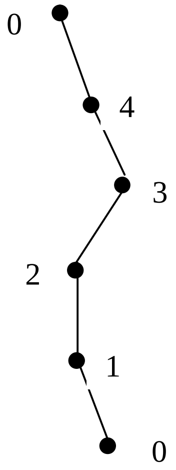
(b) ({1, 2, 4, 6, 8}, |)

Solution:

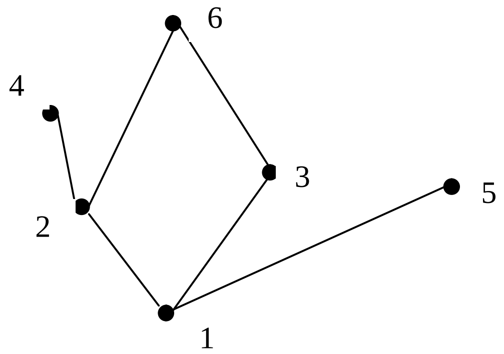
(a)	(i)	{0}and {1}	(ii)	{0, 1} and {1, 2}
(b)	(i)	4, 6	(ii)	6, 8

Problem 4.5: Draw Hasse diagrams for the following relations.

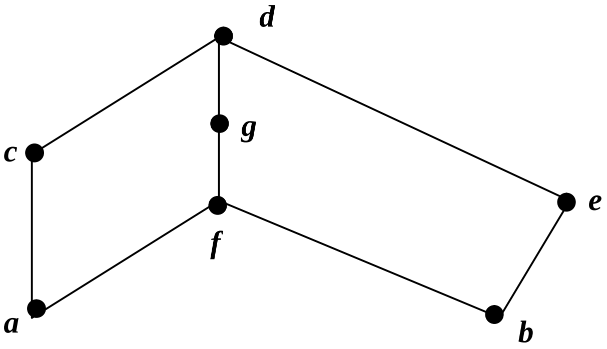
(i) ({0, 1, 2, 3, 4, 5}, ≥) (Note that it forms a chain)



(ii) ({1, 2, 3, 4, 5, 6}, |)



Problem 4.6: Determine whether the poset represented by the following Hasse diagrams, is a lattice. Justify your answer.



Solution: Given poset is a lattice, as every pair of elements has a unique least upper bound and unique greatest lower bound.

Now it is the time to check the understanding of the partial order relation.

Exercise:

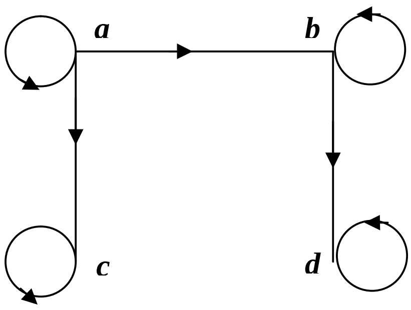
1. Define following terms with a suitable example in each of the following case.
   (i) Partial ordering relation (Apr. 04)
   (ii) Comparable elements
   (iii) Total ordering relation
   (iv) Hasse Diagram (Apr. 04)

2. Which of these relations on {0, 1, 2, 3} are partial ordering? Determine the properties of a partial ordering that the others lack.

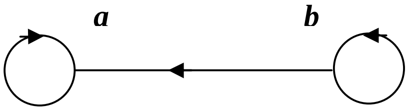
   (i) {(0, 0), (1, 1), (2, 2), (3, 3)}
   (ii) {(0, 0), (1, 1), (2, 0), (2, 2), (2, 3), (3, 2), (3, 3)}
   (iii) {(0, 0), (1, 1), (1, 2), (1, 3), (2, 0), (2, 2), (2, 3), (3, 0), (3, 3)}
   (iv) {(0, 0), (0, 1), (0, 2), (0, 3), (1, 0), (1, 1), (1, 2), (1, 3), (2, 0), (2, 2), (3, 3)}
   (v) {(0, 0), (0, 1), (1, 2), (0, 2), (1, 1), (1, 2), (2, 3), (1, 3)}
3. Determine whether the following relations, represented by a relation matrix or a diagraph, are partial ordering relations. Justify your answer.

(i) $\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$ (ii) $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

(iii)

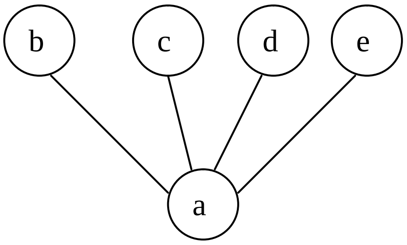


(iv)



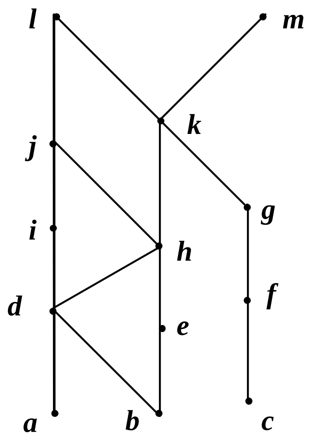
4. Draw the Hasse diagram for each of the following relations.
   - (i) P = {0, 1, 2, 3, 4, 5} and relation R: greater than or equal to
   - (ii) P = {1, 2, 3, 4, 5, 6, 7, 8} and relation R: divisibility
   - (iii) P = {1, 2, 3, 6, 12, 24, 36, 48} and relation R: divisibility
   - (iv) P = Power set of S, where S = {a, b, c, d} and relation R: $\subseteq$

5. Find matrix of partial order whose Hasse diagram is



6. With reference to partial ordering relation, define following terms with a suitable example for each of them.
   (i) Cover (ii) Upper Bound (iii) Least Upper Bound (iv) Lower bound (v) Greatest lower bound

7. With reference to the Hasse diagram (Fig. 1.1), answer the following questions.



   - (i) Find the maximal elements.
   - (ii) Find the minimal elements.
   - (iii) Is there a greatest element?
   - (iv) Is there a least element?
   - (v) Find all upper bounds of {a, b, c}.
   - (vi) Find least upper bound of {a, b, c}, if exists.
   - (vii) Find all lower bounds of {f, g, h}.
   - (viii) Find greatest lower bound of {f, g, h}, if exists.

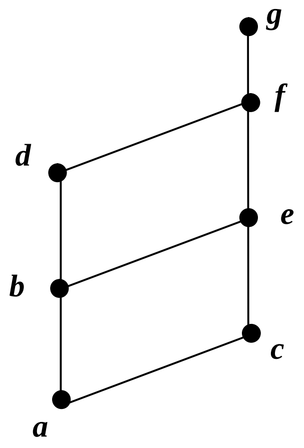
8. Define a Lattice and illustrate with a suitable example.

9. Determine whether the following are lattices. Justify your answer.

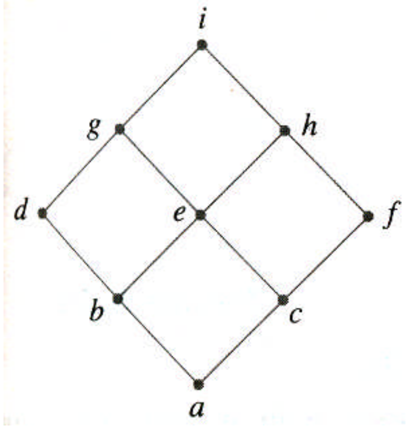
   (i) P = {1, 3, 6, 9, 12}; relation R: Divisibility
   (ii) P: Set of all divisors of 70; relation R: Divisibility
   (iii) (Z, $\geq$)

10. Determine whether the posets represented by the following Hasse diagrams, are lattices. Justify your answer.

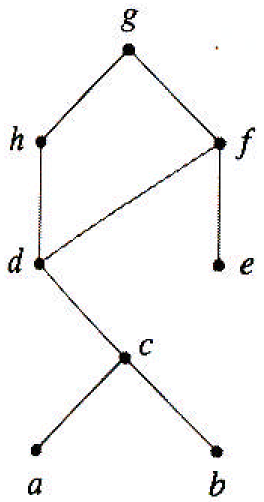
    (i)



(ii)



(iii)



11. Prerequisites in the college for various subjects are one of the partial ordering relations. We say A << B, if course A is a prerequisite of course B. Consider the mathematics courses and their prerequisites given below and draw a Hasse diagram based on it. Decide whether given relation is Lattice.

Course	Prerequisite
Math 101	None
Math 201	Math 101
Math 250	Math 101
Math 251	Math 250
Math 340	Math 201
Math 341	Math 340
Math 450	Math 101, Math 250
Math 500	Math 450, Math 251

❖❖❖❖

5

RECURRENCE RELATION

Unit Structure

5.0 OBJECTIVES:

1. Definition and examples of recurrence relation.
2. Formulation of recurrence relation.
3. Solving recurrence relations using backtracking method.
4. Solving homogeneous linear recurrence relation
5. Solving non-homogeneous linear recurrence relation

5.1 INTRODUCTION:

We are familiar with some problem solving techniques for counting, such as principles for addition, multiplication, permutations, combinations etc. But there are some problems which cannot be solved or very tedious to solve, using these techniques. In some such problems, the problems can be represented in the form of some relation and can be solved accordingly. We shall discuss some such examples before proceeding further.

Example5.1: The number of bacteria, double every hour, then what will be the population of the bacteria after 10 hours? Here we can represent number of bacteria at the n^{th} hour be a_n. Then, we can say that $a_n = 2a_{n-1}$.

Example 5.2: Our usual compound interest problems are examples of such representation. That is, $I_n = P\left(1+\frac{r}{100}\right)^n - P$, where P is principal, r is rate of interest, n is period in years and I_n is interest at the end of n^{th} year.

Example 5.3: ***Towers of Hanoi*** is a popular puzzle. There are three pegs mounted on a board, together with disks of different sizes. Initially, these discs are placed on the first peg in order of different sizes, with the largest disc at the bottom and the smallest at the top. The task is to move the discs from the first peg to the third peg using the middle peg as auxiliary. The rules of the puzzle are:

- Only one disc can be moved at a time.
- No disc can be placed on the top of a smaller disc.

This is a popular puzzle and we shall discuss its solution, using the one of the techniques discussed in this chapter.
With these illustrations, we define recurrence relation now.

Definition5.1.1: A recurrence relation for the sequence $\{a_n\}$ is an equation, that expresses a_n in terms of one or more of the previous terms of the sequence, namely, a_0, a_1, ..., a_{n-1}, for all integers n with $n \geq n_0$, where n_0 is a nonnegative integer.

Example5.4: $a_n = 1.06a_{n-1}$, with $a_0 = 0.5$.

Example 5.5: $a_n = 2a_{n-1} + 5$, with $a_0 = 1$.
The term a_0, given in the above two examples, specify **initial condition** to solve the recurrence relation completely.

5.2 FORMULATION OF RECURRENCE RELATION:

Before we proceed with discussing various methods of solving recurrence relation, we shall formulate some recurrence relation. The first example of formulation that we discuss is the problem of Tower of Hanoi that is Example 5.3 above.

Example 5.6: With reference to Example 5.3, let H_n denote the number of moves required to solve the puzzle with n discs. Let us define H_n recursively.

Solution: Clearly, $H_1 = 1$.
Consider top (n–1) discs. We can move these discs to the middle peg using H_{n-1} moves. The nth disc on the first peg can then moved to the third peg. Finally, (n–1) discs from the middle peg can be moved to the third peg with first peg as auxiliary in H_{n-1} moves. Thus, total number of moves needed to move n discs are: $H_n = 2H_{n-1} + 1$. Hence the recurrence relation for the Tower of Hanoi is:

$H_n = 1$ if $n = 1$.
$H_n = 2H_{n-1} + 1$ otherwise.

Example5.7: Find recurrence relation and initial condition for the number of bit strings of length n that do not have two consecutive 0s.

Solution: Let a_n denote the number of bit strings of length n that do not contain two consecutive 0s. Number of bit strings of length one that follow the necessary rule are: string 0 and string 1. Thus, $a_1 = 2$. The number of bit strings of length 2 is: string 01, 10 and 11. Thus, $a_2 = 3$. Now we shall consider the case $n \geq 3$. The bit strings of length n that do not have two consecutive 0s are precisely those strings length $n–1$ with no consecutive 0s along with a 1 added 1 at the end of it (which is a_{n-1} in number) and bit strings of length $n–2$ with no consecutive 0s with a 10 added at the end of it (which is a_{n-2} in number). Thus, the recurrence relation is:

$a_n = a_{n-1} + a_{n-2}$ for $n \geq 3$ with $a_1 = 2$ and $a_2 = 3$.

5.3 METHODS OF SOLVING RECURRENCE RELATION:

Now, in this section we shall discuss a few methods of solving recurrence relation and hence solve the relations that we have formulated in the previous section.

5.3.1 Backtracking Method:

This is the most intuitive way of solving a recurrence relation. In this method, we substitute for every term in the sequence in the form of previous term (i.e. a_n in the form of a_{n-1}, a_{n-1} in the form of a_{n-2} and so on) till we reach the initial condition and then substitute for the initial condition. To understand this better, we shall solve the recurrence relations that we have come across earlier.

Example5.8: Solve the recurrence relation in Example 5.4.

Solution: Given recurrence relation is $a_n = 1.06a_{n-1}$, with $a_0 = 0.5$. From this equation, we have $a_n = 1.06a_{n-1} = 1.06\times1.06\ a_{n-2} = 1.06\times1.06\times1.06\ a_{n-3}$

Proceeding this way, we have $a_n = (1.06)^n a_0$. But, we know that $a_0 = 0.5$. Thus, explicit solution to the given recurrence relation is $a_n = 0.5\times(1.06)^n$ for $n \geq 0$.

Example 5.9: Solve the Tower of Hanoi puzzle, using backtracking method.

Solution: The recurrence relation, for the puzzle is:

$H_n = 1$ if $n = 1$.

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

Thus, $H_n = 2H_{n-1} + 1 = H_n = 2 \times (2H_{n-2} + 1) + 1 = 2^2H_{n-2} + 2 + 1 = 2^2(2H_{n-3} + 1) + 2 + 1$

$= 2^3H_{n-3} + 2^2 + 2 + 1$. Proceeding this way, we have

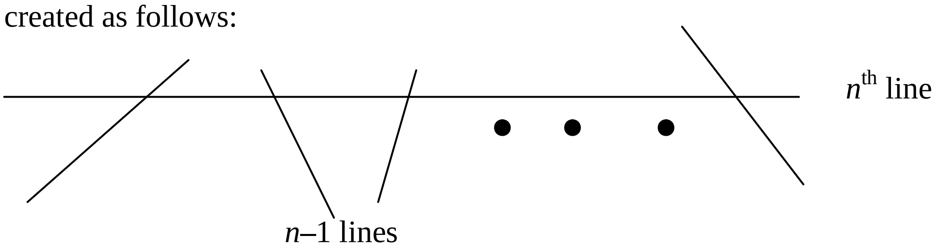
$H_n = 2^{n-1}H_1 + 2^{n-2} + 2^{n-3} + 2^{n-4} + ... + 1.$

$= 2^{n-1} + 2^{n-2} + 2^{n-3} + 2^{n-4} + ... + 1 \qquad (H_1 = 1)$

$= 2^n - 1.$

Example 5.10: Find the recurrence relation to count the number of regions created by *n* lines in a plane, such that each pair of lines has exactly one point of intersection. Solve this recurrence relation.

Solution: Let r_n be the number of regions created by n lines following the condition mentioned in the example. If the number of lines is 1, then obviously, $r_1 = 2$. If number of lines is 2, then $r_2 = 4$. Now, we shall assume that there are n–1 lines satisfying the condition mentioned. Then the number of regions created by these lines is r_{n-1}. If we add one more line, that interest each of these line exactly once then n more regions are created as follows:



Then, as we observe from above diagram, if n^{th} line intersects all n–1 lines, then new n regions are created. Thus, the recurrence relation is:

$r_n = r_{n-1} + n$, with $r_1 = 2$.

To solve this equation, we shall use the backtracking method.

$r_n = r_{n-1} + n = (r_{n-1} + n - 1) + n = ... = r_1 + 2 + 3 + ...+ n$

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

5.3.2 Method for solving linear homogeneous recurrence relations with constant coefficients:

In the previous subsection, we have seen a backtracking method for solving recurrence relation. However, not all the equations can be solved easily using this method (such as Example 5.7). In this subsection, we shall discuss the method of solving a type of recurrence relation called linear homogeneous recurrence relation. Before that we shall define this class of recurrence relation.

Definition 5.3.1: A ***linear homogeneous recurrence relation of degree k*** with constant coefficients is a recurrence relation of the form: $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$, where c_1, c_2, ..., c_k are constant real numbers with $c_k \neq 0$.

Example 5.11: Example 5.7 is a linear homogeneous recurrence relation of degree 2.

Example 5.12: Fibonacci sequence is also an example of a linear homogeneous recurrence relation of degree 2.

Example5.13: The recurrence relation $a_n = a_{n-1} + a_{n-2}^2$ is not linear (due to square term), whereas the relation $H_n = 2H_{n-1} + 1$ is not homogeneous (due to constant 1).

The basic approach for solving a linear homogeneous recurrence relation to look for the solution of the form $a_n = r^n$, where r is constant. Note that, r^n is a solution to the linear homogeneous recurrence relation of degree k, if and only if;

$r^n = c_1 r^{n-1} + c_2 r^{n-2} + \cdots + c_k r^{n-k}$. When both the sides of the equation are divided by r^{n-k} and right side is subtracted from the left side, we obtain an equation, known as characteristic equation of the recurrence relation as follows:

$r^k - c_1 r^{k-1} - c_2 r^{k-2} - \cdots - c_{k-1} r - c_k = 0$.

The solutions of the equation are called as characteristic roots of the recurrence relation.

In this subsection, we shall focus on solving linear homogeneous recurrence relation of degree 2 that is: $a_n = c_1 a_{n-1} + c_2 a_{n-2}$.
The characteristic equation of this relation is $r^2 - c_1 r - c_2 = 0$. This is a quadratic equation and has two roots. Two cases arise.

(i) Roots are distinct, say s_1 and s_2. Then, it can be shown that $a_n = us_1^n + vs_2^n$ is a solution to the recurrence relation, with $a_1 = us_1 + vs_2$ and $a_2 = us_1^2 + vs_2^2$.

(ii) Roots are equal, say s. Then it can be shown that $a_n = (u + vn)s^n$ is a solution to the recurrence relation.

We shall use above results to solve some problems.

Example 5.14: Solve the recurrence relation $b_n + 3b_{n-1} + 2b_{n-2} = 0$, with $b_1 = -2$ and $b_2 = 4$.

Solution: The characteristic equation to the given recurrence relation is $x^2 + 3x + 2 = 0$. Roots of this equation are $s_1 = -2$ and $s_2 = -1$. Hence the solution to the relation is:

$b_n = u(-1)^n + v(-2)^n$. $b_1 = -2 = -u - 2v$ and $b_2 = 4 = u + 4v$. Solving these two equations simultaneously, we get, $u = 0$ and $v = 1$. Thus, explicit solution to the given recurrence relation is $b_n = (-2)^n$

5.3.3 Method for solving linear non-homogeneous recurrence relations with constant coefficients:

In the previous subsection, we have seen a way of solving linear homogeneous recurrence relation. In this subsection, we shall discuss method of solving linear non-homogeneous recurrence relation with constant coefficient, i.e. relation of the form:
$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n)$, where $F(n)$ is a function of n and not equal to zero.

The equation, $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$, is called associated homogeneous recurrence relation.

Example 5.15: Equations $a_n = a_{n-1} + 2^n$, $a_n = a_{n-1} + a_{n-2} + n^2 + n + 1$, $a_n = 3a_{n-1} + n3^n$ and $a_n = a_{n-1} + a_{n-2} + a_{n-3} + n!$, are examples of linear non-homogeneous recurrence relations with constant coefficients and $a_n = a_{n-1},$ $a_n = a_{n-1} + a_{n-2},$ $a_n = 3a_{n-1}$ and $a_n = a_{n-1} + a_{n-2} + a_{n-3},$ are associated linear homogeneous recurrence relations respectively.

The key fact about linear non-homogeneous recurrence relations with constant coefficient is that every solution is the sum of a particular solution and a solution associated linear homogeneous recurrence relation. Thus, to put it shortly

If $\{a_n^{(p)}\}$ is a particular solution of the non-homogeneous linear recurrence relation with constant coefficients, $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n)$, then every solution of the form $\{a_n^{(p)} + a_n^{(h)}\}$, where $\{a_n^{(h)}\}$ is a solution of associated homogeneous recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$.

Though, there are no hard and fast rules for finding particular solution, depending upon the $F(n)$, there are certain guidelines for choosing a particular solution form and hence finding a particular solution. These can be understood from the following theorem.

Theorem: Suppose $\{a_n\}$ satisfies the linear non-homogeneous recurrence relation
$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n)$, where $c_1, c_2, ..., c_k$ are real numbers and
$F(n) = (b_0 + b_1 n + \cdots + b_{t-1} n^{t-1} + b_t n^t) s^n$, where $b_0, b_1, ..., b_t$ and s are real numbers. When s is not a root of the characteristic equation of the associated homogeneous recurrence relation, there is a particular solution of the form: $(p_0 + p_1 n + \cdots + p_{t-1} n^{t-1} + p_t n^t) s^n$
When s is a root of the characteristic equation and its multiplicity is m, there is a particular solution of the form:
$n^m (p_0 + p_1 n + \cdots + p_{t-1} n^{t-1} + p_t n^t) s^n$

Example 5.16: Solve the recurrence relation $a_n - 7a_{n-1} + 10a_{n-2} = 3^n$, with $a_0 = 0$ and $a_1 = 1$.

Solution: Associated homogeneous relation is: $a_n - 7a_{n-1} + 10a_{n-2} = 0$. Characteristic equation for this relation is $x^2 - 7x + 10 = 0$. Roots are: 2, 5 and hence solution to the homogeneous equation is $a_n = u2^n + v5^n$. $a_0 = O = u + v$ and $a_1 = 1 = 2u + 5v$ On solving these equations simultaneously, we get, $u = \frac{1}{3}$ and $v = \frac{-1}{3}$. Thus, solution to the associated relation is: $a_n^{(h)} = \frac{1}{3}\left(2^n - 5^n\right)$. From the given table, particular solution is of the form $p3^n$ and hence we have to determine the value of p. Hence, we have, $3^n - 7.3^{n-1} + 10.3^{n-2} = p3^n$

Solving for p, we get $p = \frac{4}{3}$ Thus, particular solution is

$a_n^{(p)} = \frac{4}{3}3^n = 4.3^{n-1}$. Solution to the given recurrence relation is

$a_n^{(h)} + a_n^{(p)} = \frac{1}{3}\left(2^n - 5^n\right) + 4.3^{n-1}$.

Example 5.17: Find all the solutions of the recurrence relation $a_n = 3a_{n-1} + 2n$. What is its solution when $a_1 = 1$?

Solution: Associated homogeneous equation is $a_n = 3a_{n-1}$ and its characteristic equation is $x^2 - 3x = 0$. The roots are 0 and 3 and hence solution is $a_n = uo^n + u3^n = v3^n$. Thus, $a_n^{(h)} = v3^n$.

Now, we shall find its particular solution. As $F(n)$ is a polynomial of degree 1, particular solution is of the form $pn + q$. Hence the recurrence relation becomes $pn + q = 3a_{n-1} + 2n$.

That is, $pn + q = 3[p(n–1) + q] + 2n$. Or $pn + q = 3pn – 3p + 3q + 2n$, i.e. $2pn + 2q – 3p = –2n$. Or $2p = –2$, $p = –1$, $2q – 3p = 0$ i.e. $2q = –3$ $q = –3/2$. Thus, we have,

$a_n^{(p)} = -n - \frac{3}{2}$. Hence, solution to given relation is: $v.3^n - n - \frac{3}{2}$. To find solution, if $a_1 = 1$, we substitute $n = 1$, in its solution. Thus, $a_1 = 1 = v.3^1 - 1 - \frac{3}{2}$. This gives $v = \frac{7}{2}$.

Hence solution to given recurrence relation is $a_n = \left(\frac{7}{2}\right)3^n - n - \frac{3}{2}$.

Example 5.18: What form does a particular solution of the linear non-homogeneous recurrence relation $a_n = 6a_{n-1} - 9a_{n-2} + F(n)$ have, when, $F(n) = 3^n$, $F(n) = n3^n$, $F(n) = n^2 2^n$ and $F(n) = (n^2+1)3^n$.

Solution: The associated linear homogeneous recurrence relation is, $a_n = 6a_{n-1} - 9a_{n-2}$.

Its characteristic equation is $x^2 – 6x + 9 = 0$. The roots are 3,3. Hence solution is $a_n = (u + vn)3^n$. To apply previous theorem, we should check the function $F(n)$.

For, $F(n) = n^2 2^n$, root is 3 and 2 is not a root and hence particular solution is of the form:

$(p_0 + p_1n + p_2n^2)3^n$. In rest of the cases we have to consider the multiplicity of the root. Thus, for $F(n) = 3^n$, particular solution is of the form: $pn^2 3^n$. For $F(n) = n3^n$, particular solution is of the form: $n^2(p_0 + p_1n)3^n$. For $F(n) = (n^2+1)3^n$, particular solution is of the form: $n^2(p_0 + p_1n + p_2n^2)3^n$.

5.4 UNIT END EXERCISE:

1. Hemant deposits Rs. 10,000 in a saving account at bank. The annual interest rate of bank is 9% that is compounded. Define a recurrence relation to compute the amount A_n his account at the end of nth year assuming that he does not withdraw money in between.

2. Let $T(n)$ denote the time required to search among n elements. Assume that n is power of 2. Let $T(n) = T(n/2)$ if $n \geq 2$ and $T(1) = 1$. Find explicit formula for $T(n)$.

3. Solve the following recurrence relation (known as 'handshake' problem):

$$H_n = H_{n-1} + (n - 1),\ \ n \geq 2, \text{ and }\ H_1 = 0.$$

4. Solve the homogeneous recurrence relation $t_n = 5t_{n-1} - 6t_{n-2}$, subject to the initial conditions $t_0 = 7$ and $t_1 = 16$.

5. Let A = { 0, 1}. Formulate recurrence relation to count number of strings that do not contain a sequence 111.

6. Solve following non-homogeneous recurrence relations
   (i) $a_n - 8a_{n-1} + 15a_{n-2} = 3^n$ with $a_0 = 0$, $a_1 = 1$.
   (ii) $a_n = 2a_{n-1} + 3.2^n$
   (iii) $a_n = 2a_{n-1} + n + 5$ with $a_0 = 4$.

6

GROUPS AND APPLICATION

[Syllabus Groups and Applications : Monoids, Semigroups, Product and quotients of algebraic structures, Isomerism, homomorphism, automorphism, Normal subgroups]

Unit Structure

7.0 OBJECTIVES:

To present the concepts of :

- Group, semigroup, products & quotients of semigroups.
- Hornomorphism, Isornorphism & automorphism of semigroups, monoids & Groups.
- Coset & Normal subgroup.

7.1 INTRODUCTION:

In this chapter, we will study, binary operation as a function, and two more algebraic structures, semigroups and groups. They are called an algebraic structure because the operations on the set define a structure on the elements of that set. We also define the notion of a hornomorphism and product and quotients of groups and semigroup.

6.2 BINARY OPERATION

A binary operation on a set A is an everywhere defined function $f : A\times A\rightarrow A$ Generally operation is defined by $* If *$ is binary operation on A then $a*b\in A \,\forall a,b\in A$

Properties of binary operation : - Let $*$ be a binary operation on a set A, Then $*$ satisfies the following properties for any a, b and c in A

1. $a=a*a$ Identity property
2. $a*b=b*a$ Commutative property
3. $a*(b*c)=(a*b)*c$ Associative property

6.3 SEMIGROUP

A non-empty set S together with a binary operation $*$ is called as a semigroup if –

i) binary operation $*$ is closed

ii) binary operation $*$ is associative

we denote the semigroup by (S, $*$)

Commutative Semigroup :- A semigroup (S, $*$) is said to be commutative if $*$ is commutative i.e. $a*b=b*a \quad \forall a\in S$

Examples : 1) (z, +) is a commutative semigroup

2) The set P(S), where S is a set, together with operation of union is a commutative semigroup.

3) (Z, –) is not a semigroup

The operation subtraction is not associative

6.4 IDENTITY ELEMENT :

An element e of a semigroup (S, $*$) is called an identity element if $e*a=a*e=a \qquad \forall a\in S$

Monoid A non-empty set M together with a binary operation *defined on it, is called as a monoid if –

i) binary operation $*$ is closed

ii) binary operation $*$ is associative and

iii) (M, $*$) has an identity.

i.e. A monoid is a semi group that has an identity

6.5 GROUP

A a non-empty set G together with a binary operation $*$ defined on it is called a group if

(i) binary operation $*$ is close,

(ii) binary operation $*$ is associative,

(iii) (G, $*$) has an identity,

(iv) every element in G has inverse in G,

We denote the group by (G, $*$)

Commutative (Abelian Group : A group (G, $*$) is said to be commutative if $*$ is commutative. i.e. $a * b = b * a \ \forall a, b \in G$.

Cyclic Group : If every element of a group can be expressed as some powers of an element of the group, then that group is called as cyclic group.

The element is called as generator of the group.

If G is a group and a is its generator then we write $G = < a >$

For example consider $G = \{1, -1, i, -i\}$. G is a group under the binary operation of multiplication. Note that $G = < i >$. Because $a = \left\{i, i^2, i^3, i^4\right\} = \left\{i, -1, -i, 1\right\}$

6.6 SUBSEMIGROUP :

Let (S, $*$) be a semigroup and let T be a subset of S. If T is closed under operation $*$, then (T, $*$) is called a subsemigroup of (S, $*$).

Submonoid : Let (S, $*$) be a monoid with identity e, and let T be a non-empty subset of S. If T is closed under the operation $*$ and e $\in$ T, then (T, $*$) is called a submonoid of (S, $*$).

Subgroup : Let (G, $*$) be a group. A subset H of G is called as subgroup of G if (H, $*$) itself is a group.

Necessary and Sufficient Condition for subgroup : Let (G; $*$) be a group. A subset H of G is a subgroup of G if and only if $a * b^{-1} \in H$ $\forall a, b \in H$

6.7 PERMUTATION

Definition : A permutation on n symbols is a bijective function of the set $A = \{1, 2, ...n\}$ onto itself. The set of all permutations on n symbols is denoted by S$_n$. If α is a permutation on n symbols, then α is completely determined by its values $\alpha(1), \alpha(2) .....\alpha(n)$. We use following notation to denote $\alpha \begin{pmatrix} 1 & 2 & 3 ..... & n \\ \alpha(1) & \alpha(1) & \alpha(3) ..... & \alpha(n) \end{pmatrix}$.

For example $\alpha \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 3 & 1 & 2 & 4 \end{pmatrix}$ denotes the permutation on the 5 symbols (1,2,3,4,5). α maps 1 to 5, 2 to 3, 3 to 1, 4 to 2 and 5 to 4.

Product of permutation : - Let A = {1,2,3,4}

Let $\alpha \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 4 & 1 \end{pmatrix}$ and $\beta \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}$.

Then $\alpha \mathrm{O} \beta = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 4 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 \end{pmatrix}$

Cycle - an element $\alpha \in s_n$ is called a cycle of length r if $\exists$ r symbols $i_1, i_2 ....i_n \alpha(i_1) = i_2, \alpha(i_2) = i_3 ... \alpha(i_n) = i_1$.

Example : Consider following permutation

i) $\alpha \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 4 & 1 & 6 & 5 \end{pmatrix}$. It can be expressed as a product of cycles -

$\alpha \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix} \begin{pmatrix} 5 & 6 \\ 6 & 5 \end{pmatrix} = (1 \; 2 \; 3 \; 4)(5 \; 6)$

Transposition :

A cycle of length two is called transposition.

For example following permutation can be expressed as a product of transpositions.

$\alpha (1837)(25)(46)$

$\therefore \alpha (1 \; 8)(1 \; 3)(1 \; 7)(25)(46)$

Even (odd) Permutation -

Let A {1, 2, ….n). A permutation $\alpha \in s_n$ is even or odd according to whether it can be expressed as the product of an even number of transpositions or the product of an odd number of transpositions respectively.

For example we can consider following permutation :

$\alpha = (1\ 4\ 5)(2\ 3)$

$\alpha = (1\ 4)(1\ 5)(2\ 3)$

= odd no. of transpositions so α is odd permutation

Example 1 : Show that $*$ defined as $x * y = x$ is a binary operation on the set of positive integers. Show that $*$ is not commutative but is associative.

Solution : Consider two positive integers x and y. By definition $x * y = x$ which is a positive integer. Hence · is a binary operation.

For commutativity : $x * y = x$ and $y * x = x$. Hence $x * y \neq y * x$ in general $\therefore$ $*$ is not commutative.

But $x*(y*z) = x*y = x$ and $(x*y)*z = x*z = x$. Hence $x*(y*z) = (x*y)*z$. $\therefore$ $*$ is associative

Example 2 : Let I be the set of integers and Z_m be the set of equivalence classes generated by the equivalence relation "congruent modulo m" for any positive integer m.

a) Write the sets Z_3 and Z_6

b) Show that the algebraic systems (Z_m, $+_m$) and (Z_m, $\times_m$) are monoids.

c) Find the inverses of elements in Z_3 and Z_4 with respect to $+_3$ and $\times_4$ respectively.

Solution : a) Z_3 for (Z_3,$+_3$) ={[0], [1], [2]}

Z_6 for (Z_6, $+_6$) = {[0], [1], [2], [3], [4], [5] }

Z_3 for (Z_3,$\times_3$) ={[0], [1], [2]}

Z_6 for (Z_6,$\times_6$) = {[0], [1], [2], [3], [4], [5] }

Example 3 : Determine whether the following set together with the binary operation is a semigroup, a monoid or neither. If it is a monoid, specify the identity. If it is a semigroup or a monoid determine whether it is commutative.

i) A = set of all positive integers. $a*b=\max\{a,b\}$ i.e. bigger of a and b [May-06]

ii) Set S = {1, 2, 3, 6, 12} where $a*b=G.C.D.(a,b)$ [Dec-03, May – 07]

iii) Set S ={1,2,3,6,9,18) where $a*b=L.C.M.\left(a,b\right)$ [Nov-06]

iv) Z, the set of integers, where $a*b=a+b-ab$ [April - 04]

v) The set of even integers E, where $a*b=\frac{ab}{2}$ [May-03]

vi) Set of real numbers with $a*b=a+b+2$

vii) The set of all m×n matrices under the operation of addition.

Solution :

i) A = set of all positive integers. $a*b=\max\{a,b\}$ i.e. bigger of a and b.

Closure Property: Since Max {a, b} is either a or b $\therefore$ $a*b\in A$. Hence closure property is verified.

Associative Property :

Since $a*(b*c)=\max\{\{a,b\},c\}=\max\ \{a,b,c\}$

= Max{a,{b, c} } = (a.b).c

$\therefore$ $*$ is associative.

$\therefore$ (A, $*$) is a semigroup.

Existence of identity : 1 $\in$ A is the identity because

1.a = Max{ 1,a}= a $\forall$ a$\in$A

$\therefore$ (A, $*$) is a monoid.

Commutative property : Since Max{a, b) = max{b, a) we have $a*b=b*a$ Hence $*$ is commutative.

Therefore A is commutative monoid.

ii) Set S = { 1,2,3,6,12} where $a*b=G.C.D.\ (a,b)$

*	1	2	3	6	12
1	1	1	1	1	1
2	1	2	1	2	2
3	1	1	3	3	3
6	1	2	3	6	6
12	1	2	3	6	12

Closure Property : Since all the elements of the table $\in$ S, closure property is satisfied.

Associative Property : Since

$a*(b*c) = a*(b*c) = a*GCD\{b,c\} = GCD\ \{a,b,c\}$

And $(a*b)*c = GCD\{a,b\}*c = GCD\{a,b,c\}$

$\therefore$ $a*(b*c) = (a*b)*c$

$\therefore$ $*$ is associative.

$\therefore$ (S, $*$) is a semigroup.

Existence of identity: From the table we observe that 12 $\in$ S is the identity

$\therefore$ (S, $*$) is a monoid.

Commutative property : Since GCD{a,b}= GCD{b,a) we have $a*b = b*a$. Hence $*$ is commutative.

Therefore A is commutative monoid

(iii) Set S ={ 1,2,3,6,9, 18} where $a*b$ =L.C.M. (a,b)

*	1	2	3	6	9	18
1	1	2	3	6	9	18
2	2	2	6	6	18	18
3	3	6	3	6	9	18
6	6	6	6	6	18	18
9	9	18	9	18	9	18
18	18	18	18	18	18	18

Closure Property : Since all the elements of the table $\in$ S, closure property is satisfied.

Associative Property : Since $a*(b*c) = a*LCM\{b,c\} = LCM\{a,b,c\}$

And $(a*b)*c = LCM\{a,b\}*c = LCM\{a,b,c\}$

$\therefore$ $a*(b*c) = (a*b)*c$

$\therefore$ $*$ is associative.

$\therefore$ (S, $*$) is a semigroup.

Existence of identity : From the table we observe that 1 $\in$ S is the identity.

$\therefore$ (S, $*$) is a monoid.

Commutative property : Since LCM{a, b} = LCM{b, a} we have $a*b = b*a$. Hence $*$ is commutative.

Therefore A is commutative monoid.

(iv) Z, the set of integers where - a * b = a + b - ab

Closure Property : - $a, b \in z$ then $a + b - ab \in z \ \forall a, b$ so * is closure.

Associate Property : Consider $a, b \in z$

$$\left(a * b\right) * c = \left(a + b - ab\right) * c$$
$$= a + b - ab + c - \left(a + b - ab\right)c$$
$$= a + b - ab + c - ac - bc + abc$$
$$= a + b + c - ab - ac - bc + abc \qquad \textbf{(1)}$$

$$a * \left(b * c\right) = a * \left(b + c - bc\right)$$
$$= a + b + c - bc - a\left(b + c - bc\right)$$
$$= a + b + c - bc - ab - ac + abc \qquad \textbf{(2)}$$

From 1 & 2

$$\left(a * b\right) * c = a * \left(b * c\right) \quad \forall a, b, c \in z$$

$\therefore$ * is associative

$\therefore$ (z, &) is a semigroup.

Existence of Identity : Let e be the identity element a * e = q

a + e - q.e = a

a + e - a.e = a

e (1-a) = 0

e = 0 or a = 1

But $a \neq 1$

E = 0

$\therefore$ $O \in Z$ is the identity element.

$\therefore$ (Z, *) is monoid.

Commutative property : $\forall\, a, b \in z$

a * b = a + b - ab

= b + a - ba

= b * a

$\therefore$ * is commutative

$\therefore$ (Z, *) is commutative monoid.

O∈Z is the identity

v) E = set of even integers. $a * b = \dfrac{ab}{2}$

Closure Property : Since $\frac{ab}{2}$ is even for a and b even. $\therefore\ a*b\in E$. Hence closure property is verified.

Associative Property : Since $a*(b*c)=q*\left(\frac{bc}{2}\right)=\frac{abc}{4}=\frac{ab}{2}*c=(a*b)*c$

$\therefore$ $*$ is associative. $\therefore$(E,$*$) is a semigroup.

Existence of identity : 2$\in$ E is the identity because $2*a=\frac{2a}{2}$= a $\forall$ a $\in$ E

$\therefore$(E, $*$) is a monoid.

Commutative property : Since $\frac{ab}{2}=\frac{ba}{2}$, we have $a*b=b*a$ Hence $*$ is commutative.

$\therefore$(E,*) is commutative monoid.

(vi) -2$\in$A is identity

(vii) $\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\in$ M is the identity

Example 4 : State and prove right or left cancellation property for a group.

Solution : Let (G,$*$) be a group.

(i) To prove the right cancellation law i.e. $a*b=c*b\Rightarrow a=c$

Let a, b, c$\in$G. Since G is a group, every element has inverse in G.

$\therefore$ $b^{-1}\in$ G

Consider $a*b=c*b$

Multiply both sides by b^{-1} from the right.

:. $(a*b)*b^{-1}=(c*b)*b^{-1}$

$\therefore$ $a*(b*b^{-1})=c*(b*b^{-1})$ Associative property

$\therefore$ $e*a=e*c$ $b*b^{-1}=e\in G$

$\therefore$ a = c e$\in$G is the identity

(ii) To prove the left cancellation law i.e. $a*b=c*b\Rightarrow a=c$

Let a, b, c$\in$G: Since G is a group, every element has inverse in G.

$\therefore a^{-1}\in$G

Consider $a*b=a*c$

Multiply both sides by a^{-1} from the left

$\therefore$ $a^{-1}*(a*b)=a^{-1}*(a*c)$

$\therefore$ $(a^{-1}*a)*b=(a^{-1}*a)*c$ Associative property

$\therefore$ $e*b=e*c$ $a^{-1}*a=e\in G$

$\therefore$ b = c e$\in$G is the identity

Example 5 : Prove the following results for a group G.

(i) The identity element is unique.

(ii) Each a in G has unique inverse a^{-1}

(iii) $(ab)^{-1} = b^{-1}a^{-1}$

Solution : (i) Let G be a group. Let e_1 and e_2 be two identity elements of G.

If e_1 is identity element then $e_1e_2 = e_2e_1 = e2$ ....(1)

If e_2 is identity element then $e_1e_2 = e_2e_1 = e_1$ ....(2)

$\therefore$ From (1) and (2) we get $e_1 = e_2$ i.e. identity element is unique.

(ii) Let G be a group. Let b and c be two inverses of a$\in$G.

lf b is an inverse of a then ab = ba = e..............(1)

If c is an inverse of a then ac = ca = e..............(2)

Where e $\in$ G be the identity element.

$\therefore$ From (1) and (2) we get ab = ac and ba = ca.

$\therefore$ b=c by cancellation law : i.e. inverse of a$\in$G is unique.

$\therefore$ inverse of $a \in G$ is unique.

(iii) Let G be a group. Let a, b $\in$ G.

Consider $(ab)(b^{-1}a^{-1})$

= $a(bb^{-1})a^{-1}$ Associative property

= $(ae)a^{-1}$ $bb^{-1} = e$, e$\in$G is identity

= $(ae)a^{-1}$ Associative property

= aa^{-1} ae = a

= e $aa^{-1} = e$

Similarly we can prove $(b^{-1}a^{-1})(ab) = e$.

Hence $(ab)^{-1} = b^{-1} a^{-1}$

Example 6 : Let G be a group with identity e. Show that if $a^2 = e$ for all a in G, then every element is its own inverse [Nov.-05]

Solution : Let G be a group.

Given $a^2 = e$ for all $a \in G$.

Multiply by a^{-1} we get

$a^{-1}a^2 = a^{-1} e$

$\therefore \quad a = a^{-1}$

i.e. every element is its own inverse

Example 7 : Show that if every element in a group is its own inverse, then the group must be abelian. [Dec-02] [5]

OR

Let G be a group with identity e. Show that if $a^2 = e$ for all a in G, then G is abelian. [May-05]

Solution : Let G be a group.

$\therefore$ For $a \in G$, $a^{-1} \in G$

$\therefore$ Consider $(ab)^{-1}$

$\therefore$ $(ab)^{-1} = b^{-1}a^{-1}$ reversal law of inverse.

$\therefore$ ab=ba every element is its own inverse

$\therefore$ G is abelian.

Example 8 : Let Z_n denote the set of integers (0, 1, .. , n-1). Let $\otimes$ be binary operation on Z_n such that $a \otimes b$ = the remainder of ab divided by n.

i) Construct the table for the operation $\otimes$ for n=4.

ii) Show that $(Z_n, \otimes)$ is a semi-group for any n.

iii) Is $(Z_n, \otimes)$ a group for any n? Justify your answer.

Solution : (i) Table for the operation $\otimes$ for n = 4.

$\otimes$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

(ii) To show that $(Z_n, \otimes)$ is a semi-group for any n.

Closure property : Since all the element in the table $\in \{0, 1, \ldots, n-1\}$, closure property is satisfied.

Assiciative property : Since multiplication modulo n is associative, associative property is satisfied.

$\therefore$ $(Z_n, \otimes)$ is a semi-group

(iii) $(Z_n, \otimes)$ is not a group for any n.

If n = 4, 2^{-1} does not exist ($1 \in G$ is the identity.)

Example 9 : Show that a group (G, $*$) is abelian if and only if for a, $b \in G$, $(a*b)^2 = a^2 * b^2$ [Nov-06]

Solution : Step-1 : Given (G, $*$) is a group and for a, $b \in G$, $(a*b)^2 = a^2 * b^2$. To prove that (G, $*$) is abelian.

Given $(a*b)^2 = a^2 * b^2$

$\therefore$ $(a*b)*(a*b) = (a*a)*(b*b)$

$\therefore$ $a*(b*a)*b = a*(a*b)*b$ Associative property

$\therefore$ $(b*a)*b = (a*b)*b$ Left cancellation law

$\therefore$ $b*a = a*b$ Right cancellation law

$\therefore$ (G, $*$) is abelian.

Step-2 : Assume that (G, $*$) is abelian.

To prove that a, $b \in G$, $(a*b)^2 = a^2 * b^2$

Consider $(a*b)^2$

$= (a*b)*(a*b)$

$= a*(b*a)*b$ Associative property

$= a*(a*b)*b$ G is abelian

$= (a*a)*(b*b)$ Associative property

$= a^2 * b^2$

Example 10 : If (G, $*$) be an abelian group, then for all a, $b \in G$, show that $(a*b)^n = a^n * b^n$.

Solution : Given (G, $*$) is abelian. To prove that for all a, $b \in G$, $(a*b)^n = a^n * b^n$

We will use the method of induction. Let P(n) be the property that for all a, $b \in G$;

$(a*b)^n = a^n * b^n$

Step-l : Check that $P^{(1)}$ is true.

$(a*b)^1 = a^1 * b^1$

$a*b = a*b$ Hence P(1) is true.

Step-2 : Assume P(k) is true for some k∈N

$(a*b)^k = a^k * b^k$

Step-3: Prove P(k+1) is true.

Consider $*(a*b)^{k+1}$

= $(a*b)^k * (a*b) = (a^k * b^k) * (a*b)$ using step-2

= $a^k * (b^k * a) * b$ Associative property

= $a^k * (a * b^k) * b$ G is abelian

= $(a^k * a) * (b^k * b)$ Associative property

= $a^{k+1} * b^{k+1}$ ∴ P(k+1) is true.

Hence P(n) is true for every n∈N

Example 11 : Let α=(1 2 3 4)(6 5 7) and β=(2 4 3)(7 5) be permutations of the set {1,2,3,…..,7}. Express α as product of transposition. Find whether α ∘ β is an even permutation or not. [Dec-99][5]

Solution : Let a=(1 2 3 4)(6 5 7)

∴ α = (1 4)(1 3)(1 2)(6 7)(6 5)

$$\alpha \circ \beta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 3 & 4 & 1 & 7 & 5 & 6 \end{pmatrix} \text{o} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 4 & 2 & 3 & 7 & 6 & 5 \end{pmatrix}$$

$$\therefore \alpha \circ \beta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 3 & 4 & 1 & 7 & 5 & 6 \end{pmatrix} = (1\,2)(5\,6)$$

∴α ∘ β is an even permutation.

Example 12 : Let A = { 1, 2, 3, 4, 5, 6} and P = $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 3 & 1 & 5 & 6 \end{pmatrix}$ be permutation on A

a) Write P as a product of disjoint cycles.

b) Find P^{-1}.

c) Find the smallest positive integer k such that $P^k = 1_A$.

[May-02][4]

Solution: Let P = $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 3 & 1 & 5 & 6 \end{pmatrix}$

(a) P = (1 2 4)(3)(5)(6)

(b) $PP^{-1} = 1$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 3 & 1 & 5 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix}$$

$\therefore$ $P^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 1 & 3 & 2 & 5 & 6 \end{pmatrix}$

(c) $P^2 =$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 3 & 1 & 5 & 6 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 3 & 1 & 5 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 1 & 3 & 2 & 5 & 6 \end{pmatrix}$$

$P^3 = p^2p =$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 1 & 3 & 2 & 5 & 6 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 3 & 1 & 5 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix}$$

$\therefore$ Smallest k=3

Example 13 : Consider the group G = {1,2,3,4,5,6} under multiplication modulo 7. [Apr-04, May-06]

(i) Find the multiplication table of G

(ii) Find 2^{-1}, 3^{-1}, 6^{-1}.

(iii) Find the order of the subgroups generated by 2 and 3.

(iv) Is G cyclic?

Solution : (i) Multiplication table of G

Binary operation $*$ is multiplication modulo 7.

*	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

From the table we observe that $1 \in G$ is identity.

(ii) To find 2^{-1}, 3^{-1}, 6^{-1}.

From the table we get $2^{-1} = 4$, $3^{-1} = 5$, $6^{-1} = 6$

iii) To find the order of the subgroups generated by 2.

Consider $2^\circ = 1$ = Identity, $2^1 = 2$; $2^2 = 4$, $2^3 = 1$ = Identity

$< 2 > = \{2^1, 2^2, 2^3\}$

$\therefore$ Order of the subgroup generated by 2 =3

To find the order of the subgroups generated by 3.

Consider $3^\circ = 1$ = identity, $3^1 = 3$, $3^2 = 2$, $3^3 = 6$, $3^4 = 4$, $3^5 = 5$, $3^6 = 1$ = Identity

$< 3 > = \{3^1, 3^2, 3^3, 3^4, 3^5, 3^6\}$

$\therefore$ Order of the subgroup generated by 3 = 6

(iv) G is cyclic because G = < 3 >.

Example 14 : Let S=$\{x|x$ is a real number and $x\neq 0$, $x\neq 1\}$. Consider the following functions $f_i : S\to S$, i=1,2,---,6 [Nov-05]

$$f_1(x) = x,\ f_2(x) = 1-x,\ f_3(x) = \frac{1}{x},\ f_4(x) = \frac{1}{1-x},\ f_5(x) = 1-\frac{1}{x},$$

$$f_6(x) = \frac{x}{x-1}$$

Show that G = $\{f_1, f_2, f_3, f_4, f_5, f_6)$ is a group under the operation of composition. Give the multiplication table of G.

Solution : (i) Multiplication table of G

	f_1	f_2	f_3	f_4	f_5	f_6
f_1	f_1	f_2	f_3	f_4	f_5	f_6
f_2	f_2	f_1	f_5	f_6	f_3	f_4
f_3	f_3	f_4	f_1	f_2	f_6	f_5
f_4	f_4	f_3	f_6	f_5	f_1	f_2
f_5	f_5	f_6	f_2	f_1	f_4	f_3
f_6	f_6	f_5	f_4	f_3	f_2	f_1

(i) **Closure property :** Since all the elements in the table $\in$G, closure property is satisfied.

(ii) **Associative property :** Since composition of functions is associative, associative property is satisfied.

(iii) **Existence of identity :** From the table we observe that $f_1 \in G$ is the identity.

(iv) **Existence of inverse :** From the table we observe that

$f_1^{-1} = f_1$, $f_2^{-1} = f_2$, $f_3^{-1} = f_3$, $f_4^{-1} = f_5$, $f_5^{-1} = f_4$, $f_6^{-1} = f_6$

i.e. every element of G has inverse in G. Hence G is a group.

Example 15 : Let G be an abelian group with identity e and let H = {x/x^2 = e). Show that H is a subgroup of G. [May-02, 03, May-07]

Solution : Let x, y$\in$H $\therefore x^2 = e$ and $y^2 = e$ $\therefore$ $x^{-1} = x$ and $y^{-1} = y$

Since G is abelian we have xy = yx $\therefore$ $xy^{-1} = yx$

Now $(xy^{-1})^2$ = $(xy^{-1})(xy^{-1}) = (xy^{-1})(y^{-1}x)$

= $(xy^{-1})(yx) = x(y^{-1}y)x$

= $x(e)x$

= $x^2 = e$

$\Rightarrow$ $xy^{-1} \in H$

$\therefore$ H is a subgroup.

Example 16 : Let G be a group and let H = (x/x$\in$G and xy = yx for all y$\in$G}. Prove that H is a subgroup of G. [98][7]

Solution : Let x, z $\in$ H $\therefore$ xy = yx for every y$\in$G $\therefore$ $x = yxy^{-1}$.

Similarly zy = yz for every y$\in$G $\therefore z = yzy^{-1}$.

Now consider xz^{-1} = $(yxy^{-1})(yzy^{-1})^{-1}$

= $yxy^{-1}yz^{-1}y^{-1} = yxz^{-1}y^{-1}$

$\Rightarrow$ $(x.z^{-1})y = y(xz^{-1}) \in H$.

$\Rightarrow$ $xz^{-1} \in H$

$\therefore$ H is a subgroup

Example 17 : Find all subgroups of (Z,⊕) where ⊕ is the operation addition modulo 5. Justify your answer.

Solution:

⊕	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Example 18 : Let G be a group of integers under the operation of addition. Which of the following subsets of G are subgroups of G?

(a) the set of all even integers,

(b) the set of all odd integers. Justify your answer.

Solution:

a) Let H= set of all even integers.

We know, additive inverse of an even number is even and sum of two even integers is also even. Thus for a,b$\in$H we have $ab^{-1}\in$H.

Hence H is a subgroup of G.

b) Let K = set of all odd integers.

We know, additive inverse of an odd number is odd and sum of two odd integers is even.

Thus for a,b$\in$K we have $ab^{-1}\notin$K.

Hence K is not a subgroup of G.

Example 19 : Let (G, $*$) be a group and H be a non-empty subset of G. Show that (H, $*$) is a subgroup if for any a and b in H, ab^{-1} is also in H.
[May-00) [3]

Solution :

(i) Let a, a $\in$ H $\therefore$ a a^{-1} $\in$ H. i.e. e $\in$ H

$\therefore$ The identity element $\in$ H.

(ii) Let e, a $\in$ H $\therefore$ ea^{-1} $\in$ H. i.e. a^{-1} $\in$ H

$\therefore$ Every element has inverse $\in$ H.

(iii) Let a, b $\in$ H. $\therefore$ b^{-1} $\in$ H. $\therefore$ $a(b^{-1})^{-1}$ $\in$ H. i.e. ab $\in$ H.

$\therefore$Closure property is satisfied.

(iv) Every element in H is also in G. And G is a group. So associative property is satisfied by the elements of H. Hence associative property is satisfied by the elements of H.

Hence H is a group. But H is a subset of G. $\therefore$H is a subgroup of G.

Example 20 : Let H and K be subgroups of a group G. Prove that H$\cap$K is a subgroup of G. [Dec-02] [5]

Solution : If H is a subgroups of a group G, then for any a, b $\in$ H, ab^{-1} $\in$ H.

Similarly, if K is a subgroups of a group G, then for any a, b $\in$ K, ab^{-1} $\in$ K.

Now if a, b $\in$ H$\cap$K, a, b $\in$ H and a, b $\in$ K. $\therefore$ ab^{-1} $\in$ H and ab^{-1} $\in$ K. Hence ab^{-1} $\in$ H$\cap$K.

$\therefore$ H$\cap$K is a subgroup of G.

6.8 PRODUCTS AND QUOTIENTS OF SEMIGROUPS:

In this section we obtain new semigroups from existing semigroups.

Theorem 6.1 :

If (S, $*$) and (T, $*'$) are semigroups, then (S × T, $*''$) is a semigroup, where $*''$ is defined by $(s_1,t_1) *''(s_2,t_2) = \left(s_1 * s_2, t_1 *' t_2\right)$

Theorem 6.2 :

If S and T are monoids with identities e_s and e_T, respectively, then, S × T is a monoid with identity (e_s, e_T)

Theorem 6.3 :

Let R be congruence relation on the semigroup (S, $*$). Consider the relation from S/R×S/R to S/R in which the ordered pair ([a], [b]) is, for a and b in S, related to [a $*$ b].

(a) $\otimes$ is a function from S/R×S/R to S/R, and as usual we denote $\otimes$ ([a],[b]) by [a] $*$ [b]. Thus [a] $\otimes$ [b]=[a$*$b].

(b) (S/R, $\otimes$) is a semigroup.

Proof : Suppose that ([a],[b]) = ([a'],[b']). Then aRa' and bRb', so we must have a$*$bRa'$*$b', since R is a congruence relation. Thus [a$*$b]=[a'$*$b']; that is, $\otimes$ is a function. This means that $\otimes$ is a binary operation on S/R.

Next, we must verify that $\otimes$ is an associative operation. We have

$[a]\otimes([b]\otimes[c])=[a]\otimes[b*c]=[a*(b*c)]=[(a*b)*c]$ by associative property of $*$ in S

$= [a*b] \otimes [c]$

$= ([a] \otimes [b]) \otimes [c],$

Hence S/R is a semigroup. We call S/R the ***quotient semigroup*** or ***factor semigroup***. Observe that $\otimes$ is a type of "quotient binary relation" on S/R that is constructed from the original binary relation $*$ on S by the congruence relation R

Example 21 : Let Z be the set of integers, and Z_m, be the set of eduivalences classes generated by the equivalence relation "congruence modulo m" for any positive integer m.

Z_m, is a group with operation ⊕ where [a] ⊕ [b] = [a+b]
For Z_2 and Z_3 defined according to the above definition, write the multiplication table for the group $Z_2 \times Z_3$. [May-03] [5]

Solution : The multiplication table for the group $Z_2 \times Z_3$.

⊕	(0,0)	(0,1)	(0,2)	(1,0)	(1,1,)	(1,2)
(0,0)	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)
(0,1)	(0,1)	(0,2)	(0,0)	(1,1)	(1,2)	(1,0)
(0,2)	(0,2)	(0,0)	(0,1)	(1,2)	(1,0)	(1,1)
(1,0)	(1,0)	(1,1)	(1,2)	(0,0)	(0,1)	(0,2)
(1,1)	(1,1)	(1,2)	(1,0)	(0,1)	(0,2)	(0,0)
(1,2)	(1,2)	(1,0)	(1,1)	(0,2)	(0,0)	(0,1)

6.9 HOMOMORPHISM, ISOMORPHISM AND AUTOMORPHISM OF SEMIGROUPS

Homomorphism : Let (S, $*$) and (T, $*$') be two semigroups. An everywhere defined function

f : S→T is called a homomorphism from (S, $*$) to (T, $*$') if

$f(a*b) = f(a) *'f(b) \quad \forall a, b \in S$

Isomorphism : Let (S, $*$) and (T, $*$') be two semigoups. A function

f : S → T is called a isomorphism from (S, $*$) to (T, $*$') if

(i) it is one-to-one correspondence from S to T (ii) $f(a*b) = f(a) *'f(b) \ \forall a, b \in S$

(S, $*$) and (T, $*$') are isomorphic' is denoted by $S \cong T$.

Automorphism : An isomorphism from a semigroup to itself is called an automorphism of the semigoup. An isonorptism $f:s \to s$ is called automorphism.

6.10 HOMOMORPHISM, LSOMORPHISM AND AUTOMORNHISM OF MONOIDS :

Homomorphism : Let (M, $*$) and (M', $*$') be two monoids. An everywhere defined function f : M → M' is called a homomorphism from (M, $*$) to (M', $*$') if

$f(a * b) = f(a) *'f(b) \ \forall a, b \in M$

Isomorphism : Let (M, $*$) and (M', $*$') be two monoids. A function

f : M → M' is called a isomorphism from (M, $*$) to (M', $*$') if

(i) it is one-to-one correspondence from M to M' (ii) f is onto.

(iii) f(a$*$b = f (a) $*$'f (b) $\forall$ a, b$\in$M

'(M $*$) and (M', $*$') are isomorphic is denoted by M $\cong$ M'.

Automorphism : An isomorphism from a monoid to itself is called an automorphism of the monoid. An isomorphism $f:M \to M$ is called Automorphism of monoid.

6.11 HOMOMORPHISM, ISOMORPHISM AND AUTOMORPHISM OF GROUPS :

Homomorphism : Let (G, $*$) and (G', $*$') be two groups. An everywhere defined function f : G $\to$ G' is called a homomorphism from (G, $*$) to (G', $*$') if

f (a$*$b) = f (a) $*$'f (b) $\forall$ a, b $\in$ G

Isomorphism : Let (G, $*$) and (G', $*$') be two groups. A function

f : G$\to$G' is called a isomorphism from (G, $*$) to (G', $*$') if

(i) it is one-to-one correspondence from G to G' (ii) f is onto.

(iii) f(a $*$ b) = f (a) $*$'f (b) $\forall$ a, b$\in$G

'(G, $*$) and (G', $*$') are isomorphic' is denoted by G $\cong$ G'.

Automorahism: An isomorphism from a group to itself is called an automorphism of the group. An isomorphism $f:G \to G$ is called Automorphism.

Theorem 6.4 : Let (S, $*$) and (T, $*$') be monoids with identity e and e', respectively. Let f : S $\to$ T be an isomorphism. Then f(e) = e'.

Proof : Let b be any element of T. Since f is on to, there is an element a in S such that f(a) = b

Then $a = a * e$

$$b = f(a) = f(a * e) = f(a) * f(e) = b *' f(e) \text{ (f is isomorphism)}$$

Similarly, since $a = e * a$,

$$b = f(a) = f(e * a) f(e * a) = f(e) *' (a)$$

Thus for any ,b$\in$T,

$$b = b *' f(e) = f(e) *' b$$

which means that f(e) is an identity for T.

Thus since the identity is unique, it follows that f(e)=e'

Theorem 6.5: Let (S, $*$) and (T, $*$’) be monoids with identity e and e’, respectively. Let f : S → T be a homomorphism. Then f(e) = e’.

Proof : It can be prove similarly like Theorem 6.4.

Theorem 6.6 : Let f be a homomorphism from a semigroup (S, $*$) to a semigroup (T, $*$’). If S’ is a subsemigroup of (S, $*$), then

F(S’) = {t ∈ T | t = f (s) for some s ∈ S},

The image of S’ under f, is subsemigroup of (T, $*$’).

Proof : If t_1, and t_2 are any elements of F(S’), then there exist s_1 and s_2 in S’ with

t_l= f(s_1) and t_2 = f(s_2).

Therefore,

$t_1 * t_2 = f(s_1) * f(s_2) = f(s_1 * s_2) = f(s_2 * s_1) = f(s_2) * f(s_1) = t_2 * t_1$

Hence (T, $*$') is also commutative.

Example 22 : Let G be a group. Show that the function f : G → G defined by f(a) = a^2 is a homomorphism iff G is abelian. [98][6], [May-00] [4]

Solution :

<u>Step-1 :</u> Assume G is abelian. Prove that f : G → G defined by f(a) = a^2 is a homomorphism.

Let a,b∈G. ∴ f(a) = a^2 , f(b) = b^2 and f(ab) = $(ab)^2$ by definition of f.

∴	f(ab)=$(ab)^2$		
	=	(ab)(ab).	
	=	a(ba)b	associativity
	=	a(ab)b	G is abelian
	=	(aa)(bb)	associativity
	=	a^2b^2	
	=	f(a)f(b)	definition of f

∴ f is a homomorphism.

Step 2 : $\forall\, y = a^2 \in G\ \exists\, a \in G\ s\,t$

$f(a) = y = a^2$

∴ f is onto.

<u>Step-3 :</u> Assume, f : G → G defined by f(a) = a^2 s a homomorphism. Prove that G is abelian.

Let a,b∈G. ∴ f(a) = a^2 , f(b) = b^2 and f(ab) = $(ab)^2$ by definition of f.

$\therefore$ f(ab) = f(a)f(b) f is homomorphism

$\therefore$ $(ab)^2 = a^2 b^2$ definition of f

$\therefore$ (ab)(ab) = (aa)(bb)

$\therefore$ a(ba)b = a(ab)b associativity

$\therefore$ ba = ab left and right cancellation taws

$\therefore$ G is abelian.

Example 23 : Let G be a group and let a be a fixed element of G. Show that the function $f_a : G \to G$ defined by $f_a(x) = axa^{-1}$ for x$\in$G is an isomorphism. [Dec-O2][5]

Solution :

Step-1: Show that f is 1-1.

$f_a(x) = axa^{-1}$

Consider $f_a(x) = f_a(y)$ for x, y $\in$G

$\therefore$ $axa^{-1} = aya^{-1}$ definition of f

$\therefore$ x = y left and right cancellation laws

$\therefore$ f is 1- 1

Step 2 : $\forall\, y = axa^{-1} \in G \;\exists\, x \in G$ s.t.
$f_a(x) = a\,xa^{-1}$

$\therefore$ f is onto.

Step-3 : Show that f is homomorphism.

For x, y$\in$G

$f(x) = a * x * a^{-1}$, $f(y) = a * y * a^{-1}$ and $f(x * y) = a * (x * y) * a^{-1}$

Consider $f(x * y) = a * (x * y) * a^{-1}$ for x, y$\in$G

$\therefore$ $f(x * y) = a * (x * e * y) * a^{-1}$ e$\in$G is identity

$= a * (x * a^{-1} * a * y) * a^{-1}$ $a^{-1} * a = e$

$= (a * x * a^{-1}) * (a * y * a^{-1})$ associativity

$\therefore$ $* f(x * y) = f(x) * f(y)$

$\therefore$ f is homomorphism.

Since f is 1-1 and homomorphism, it is isomorphism.

Example 24 : Let G be a group. Show that the function f : G → G defined by f(a) = a^{-1} is an isomorphism if and only if G is abelian. [May-03][4]

Solution :

Step-1: Assume G is abelian. Prove that f : G → G defined by $f(a) = a^{-1}$ is an isomorphism.

i) Let f(a)=f(b)

$\therefore a^{-1} = b^{-1}$ $\quad \therefore a = b$ $\quad \therefore$ f is 1- l.

ii) $\forall a \in G \Rightarrow a^{-1} \in G$

$\therefore x^1 \in G$

$\Rightarrow f(x) = x^{-1}$

$\therefore$ f is onto.

iii) Let $a,b \in G$. $\therefore f(a) = a^{-1}$, $f(b) = b^{-1}$ and $f(ab) = (ab)^{-1}$ by definition of f.

$\therefore$ f(ab)	=	$(ab)^{-1}$	
	=	$b^{-1}a^{-1}$	reversal law of inverse
	=	$a^{-1}b^{-1}$	G is abelian
	=	f(a)f(b)	definition of f.

$\therefore$ f is a homomorphism.

Since f is 1-1 and homomorphism, it is isomorphism.

Step – 2 : Assume f : G → G defined by $f(a) = a^{-1}$ is an isomorphism. Prove that G is abelian.

Let a, $b \in G$ $\quad \therefore f(a) = a^{-1}$, $f(b) = b^{-1}$ and $f(ab) = (ab)^{-1}$ by definition of f

$\therefore$	f(ab) = f(a)f(b)	f is homomorphism
$\therefore$	$(ab)^{-1} = a^{-1}b^{-1}$	definition of f
$\therefore$	$b^{-1}a^{-1} = a^{-1} b^{-1}$	reversal law of inverse

G is abelian.

Example 25 : Define (Z, +) → (5Z, +) as f(x) = 5x, where 5Z=(5n : n ∈ Z). Verify that f is an isomorphism. [Dec-99j [S]

Solution:

Step -1 Show that f is 1-l.

Consider	f(x) = f(y)	for x, y∈G	
$\therefore$	5x = 5y	definition of f	
$\therefore$	x = y	$\therefore$ f is 1-1	

Step 2 : $\forall 5x \in G, \exists x \in G$

s.t. $f(x) = 5x$

$\therefore$ f is onto.

Step-3: Show that f is homomorphism.

For $x * y \in G$

f(x) = 5x, d(y) = 5y and f(x+y) – 5(x+y)

Consider f(x+y) = 5(x+y) for x, y $\in$G

= 5x + 5y

$\therefore$ f(x+y) = f(x) + f(y)

$\therefore$ f is homomorphism.

Since f is 1-1 and homomorphism, it is isomorphism.

Example 26 : Let G be a group of real numbers under addition, and let G' be the group of positive numbers under multiplication. Let f : G → G' be defined by f(x) = e^x. Show that f is an isomorphism from G to G'

[May-06]

OR

Show that the group G = (R,+) is isomorphic to G' = (R^+, x) where R is the set of real numbers and R^+ is a set of positive real numbers.

Solution :

Step 1: Show that f is 1-1.

Consider f(x) = f(y) for x,y$\in$G

$\therefore$ $e^x = e^y$ definition of f

$\therefore$ x = y $\therefore$ f is 1-1.

Step 2 : If $x \in G^1$, then log $x \in G$ and $f(.\log x) = e^{\log x} = x$ so f is onto.

Step-3 : Show that f is homomnrphism.

For x, y$\in$G

f(x) = e^x, f(y) = e^y and f(x+y) = $e^{(x+y)}$

Consider f(x + y) = $e^{(x+y)}$ for x, y $\in$G

= $e^x \times e^y$

$\therefore$ f(x + y) = f(x) × f(y) f is homomorphism.

Since f is 1-1 and homomorphasm, it is isomorphism.

Example 27 : Let G = {e, a, a^2, a^3, a^4, a^5} be a group under the operation of $a^i a^i = a^r$, where i + j $\equiv$ r(mod 6). Prove that G and Z_6 are isomorphic

[May-07]

Solution :

Step - I : Show that f is l-I.

Let x = a^i, and y = a^j .

Consider f(x) = f(y) for x, y $\in$ G

$\therefore$ f(a^i) = f(a^j) definition of f

$\therefore$ $a^i = a^j$

$\therefore$ x = y f is 1-1.

Step-2 : Show that f is homomorphism.

Let x = a’ and y = a’ x, y $\in$ G

f(a^i) = i , f(a^j) j and f(x + y) = f(a^i a^j)

Consider f(x+y) = f($a^i a^j$) = f(a’) where i + j = r(mod 6)

= r

= i + j

= f(a^i) + f(a^j)

$\therefore$ f(x × y) = f(x) + f(y) $\therefore$ f is homomorphism.

Since f is 1-1 and homomorphism, it is isomorphism.

Example 28 : Let T be set of even integers. Show that the semigroups (Z, +) and (T, +) are isomorphic. [May-05]

Solution : We show that f is one to one onto .

Define f : (Z, +) $\rightarrow$ (T, +) as f(x) = 2x

1) Show that f is l-1

Consider f(x) = f(y)

$\therefore$ 2x = 2y

$\therefore$ x = y $\therefore$ f is 1-l.

2) Show that f is onto

y = 2x $\therefore$ x = y/2 when y is even.

$\therefore$ for every y$\in$T there exists x$\in$Z.

$\therefore$ f is onto.

$\therefore$ f is isomorphic.

3) F is homorphism

F (x + y) = 2 (x + y)

= 2x + 2y

= f(x) + f(y)

$\therefore$ f is honomorphism.

Example 29 : For the set A = {a,b,c} give all the permutations of A. Show that the set of all permutations of A is a group under the composition operation.

Solution : A={a,b,c}. S_3= Set of all permutations of A.

$$f_0 = \begin{pmatrix} a & b & c \\ a & b & c \end{pmatrix}, \quad f_1 = \begin{pmatrix} a & b & c \\ a & c & b \end{pmatrix}, \quad f_2 = \begin{pmatrix} a & b & c \\ c & b & a \end{pmatrix}$$

$$f_3 = \begin{pmatrix} a & b & c \\ b & a & c \end{pmatrix}, \quad f_4 = \begin{pmatrix} a & b & c \\ b & c & a \end{pmatrix}, \quad f_5 = \begin{pmatrix} a & b & c \\ c & a & b` \end{pmatrix}$$

Let us prepare the composition table.

0	f_0	f_1	f_2	f_3	f_4	f_5
f_0	f_0	f_1	f_2	f_3	f_4	f_5
f_1	f_1	f_0	f_4	f_5	f_2	f_3
f_2	f_2	f_3	f_0	f_4	f_3	f_1
f_3	f_3	f_4	f_5	f_0	f_1	f_2
f_4	f_4	f_3	f_1	f_2	f_5	f_0
f_5	f_5	f_2	f_3	f_1	f_0	f_4

i) **Closure Property:** Since all the elements in the composition table $\in S_3$, closure property is satisfied.

ii) **Associative Property:** Since composition of permutations is associative, associative property is satisfied.

iii) **Existance of Identity:** From the table we find that fo is the identity

iv) **Existance of Inverse:** From the composition table it is clear that

$f_0^{-1} = f_0, \; f_1^{-1} = f_1, \; f_2^{-1} = f_2, \; f_3^{-1} = f_3, \; f_4^{-1} = f_5, \; f_5^{-1} = f_4$

$\therefore$ Every element has inverse in S_3. Hence S_3 is a group.

6.12 COSET AND NORMAL SUBEROUP:

Left Coset : Let (H, $*$) be a subgroup of (G, $*$). For any a $\in$ G, the set of aH defined by $aH = \{a * h / h \in H\}$ is called the **left coset** of H in G determined by the element a$\in$G. The element a is called the representative element of the left coset aH.

Right Coset : Let (H, $*$) be a subgroup of (G, $*$). For any a $\in$ G, the set of Ha defined by

$Ha = [h * a \,|\, h \in H]$

is called the **right coset** of H in G determined by the element a∈G. The element a is called the representative element of the right coset Ha.

Theorem 6.7: Let (H, ∗) be a subgroup of (G, ∗). The set of left cosets of H in G form a partition of G. Every element of G belongs to one and only one left coset of H in G.

Theorem 6.8 : The order of a subgroup of a finite group divides the order of the group.

Corollary : If (G, ∗) is a finite group of order n, then for any a∈G, we must have $a^n=e$, where e is the identity of the group.

Normal Subgroup : A subgroup (H, ∗) of (G, ∗) is called a normal subgroup if for any a∈G, aH = Ha.

Example 30 : Determine all the proper subgroups of symmetric group (S_3, o). Which of these subgroups are normal?

Solution : S = {1, 2, 3}. S_3 = Set of all permutations of S.
$S_3 = \{f_0, f_1, f_2, f_3, f_4, f_5\}$ where

$$f_0 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \quad f_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \quad f_3 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

$$f_3 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}, \quad f_4 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}, \quad f_5 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

Let us prepare the composition table.

0	f_0	f_1	f_2	f_3	f_4	f_5
f_0	f_0	f_1	f_2	f_3	f_4	f_5
f_1	f_1	f_0	f_4	f_5	f_2	f_3
f_2	f_2	f_3	f_0	f_4	f_3	f_1
f_3	f_3	f_4	f_5	f_0	f_1	f_2
f_4	f_4	f_3	f_1	f_2	f_5	f_0
f_5	f_5	f_2	f_3	f_1	f_0	f_4

From the table it is clear that $\{f_0, f_1\}$, $\{f_0, f_2,\}$, $\{f_0, f_3)$ and $\{f_0, f_4, f_5\}$ are subgroups of (S_3, 0): The left cosets of $\{f_0, f_1\}$ are $\{f_0, f_1\}$, $\{f_2, f_5\}$, $\{f_3, f_4\}$. While the right cosets of $\{f_0, f_1\}$ are $\{f_0, f_1\}$, $\{f_2, f_4\}$, $\{f_3, f_5\}$. Hence $\{f_0, f_1\}$ is not a normal subgroup.

Similarly we can show that $\{f_0, f_2\}$ and $\{f_0, f_1\}$ are not normal subgroups.

On the other hand, the left and right cosets of $\{f_0, f_4, f_5\}$ are $\{f_0, f_4, f_5\}$ and $\{f_1, f_2, f_3\}$.
Hence $\{f_0, f_4, f_5\}$ is a nomal subgroup.

Example 31: Let S = {1, 2, 3}. Let G = S_3 be the group of all permutations of elements of S, under the operation of composition of permutations.

Let H be the subgroup formed by the two permutations $\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$ and $\begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$. Find the left coset of H in G. Is H a normal subgroup? Explain your notion of composition clearly. [Dec-02, Nov-06]

Solution : Let

$$f_0 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \quad f_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \quad f_3 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

$$f_3 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}, \quad f_4 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}, \quad f_5 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

$\therefore$ H=$\{f_0, f_2\}$

<u>Left Cosets of H in G :</u>

$f_0H = \{f_0f_0, f_0f_2\} = \{f_0, f_2\}$ $\qquad$ $f_1H = \{f_1f_0, f_1f_2\} = \{f_1, f_4\}$

$f_2H = \{f_2f_0, f_2f_2\} = \{f_2, f_0\}$ $\qquad$ $f_3H = \{f_3f_0, f_3f_2\} = \{f_3, f_5\}$

$f_4H = \{f_4f_0, f_4f_2\} = \{f_4, f_1\}$ $\qquad$ $f_5H = \{f_5f_0, f_5f_2\} = \{f_5, f_3\}$

<u>Right Cosets of H in G</u>

$Hf_0 = \{f_0f_0, f_2f_0\} = \{f_0, f_2\}$ $\qquad$ $Hf_1 = \{f_0f_1, f_2f_1\}=\{f_1, f_3\}$

Since $f_1 H \neq Hf_1$, H is not a normal subgroup of G.

Example 32 : Consider the dihedral group (D_4, 0). Find the subgroup of D_4 generated by $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$ Is it normal subgroup. Find the left cosets of D_4.

[Dec-99][6]

Solution: $D_4 = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$

Example 33 : Define a normal sub-group. Let S_3 = Group of all permutations of 3 elements (say 1, 2, 3). For the following subgroups of S, find all the left cosets . Subgroup of A = {1,(1,2)}

Where I = identity permutation, (1, 2) is a transposition. Is A a normal subgroup. State a normal subgroup of the above group if it exists. [98][7]

Solution : $H = \{f_0, f_3\}$

The left cosets of H in G are as follow.

$f_0H = \{f_0, f_3\}$ $f_1H = \{f_1, f_5\}$ $f_2H = \{f_2, f_4\}$

$f_3H = \{f_3, f_0\}$ $f_4H = \{f_4, f_2\}$ $f_5H = \{f_5, f_1\}$

Consider a right coset $Hf_1 = \{f_1, f_4\}$

Since $f_1H \neq Hf_1$, H is not a normal subgroup of G.

6.13 UNIT END EXERCISES

1) Determine whether the set Q, the set of all rational number with the binary operation of addition is a group. If it is a group, determine if it abelian, specify the identity and the inverse of a general element.

2) If G is a set of all not-zero real numbers and $a * b = \frac{ab}{2}$, show that (G, $*$) is an abelian group. [May-05]

3) Let G be a set of integers between 1 and 15 which are co-prime to 5. Find the multiplication table of G. Find 2^{-1}, 7^{-1}, 11^{-1}. Is G cyclic? [May-05]

4) Check whether it is an abelion group in each of the following cases-

i) R, set of real numbers where a * b = a + b +7

ii) 5 = Q × Q with operation defined as (a, b) * (c, d) = (ac, ad + b).

5) Determine whether the following sets along with the binary operation, form a group. If it is a group, state the identity, and the inverse of an element a. If it is not a group, state the reason why ?

[Oct-03]

i) Set is P(S) = set of all subsets of S where S is a non-empty set. The operation is that of union.

ii) Set of all non-zero real numbers, under the operation of multiplication.

6) Let H be a subgroup of a group G. Define the following [Oct-03]

i) Left coset of H in G.

ii) Right coset of H in G.

7) If G is a finite group then prove that $a^{|G|} = e$.

❖❖❖❖

7

CODES AND GROUP CODES

Unit Structure :

7.0 OBJECTIVES :

- To know about group code. Coding theory has developed techniques to detect and correct errors.
- To know about parity check matrix and decode words using maximum likelihood technique.

7.1 INTRODUCTION :

In today's modern world of communication, data items are constantly being transmitted from point to point.

Different devices are used for communication. The basic unit of information is message. Messages can be represented by sequence of dots and dashes.

Let $B=\left\{0,1\right\}$ be the set of bits. Every character or symbol can be represented by sequence of elements of B. Message are coded in O's and 1's and then they are transmitted. These techniques make use of group theory. We will see a brief introduction of group code in this chapter. Also we will see the detection of error in transmitted message.

The set $B=\{0,1\}$ is a group under the binary operation $\oplus$ whose table is as follows :

$\oplus$	0	1
0	0	1
1	1	0

We have seen that B is a group as the $\mathbb{Z}2$, where + is only mod 2 addition.

If follows from theorem - "If G_1 and G_2 are groups then $G=G_1\times G_2$ is a group with binary operation defined by $(a_1,b_1)(a_2,b_2)=(a_1,a_2,b_1,b_2)$. So $B^m=B\times B\times---\times B$ (m factors) is a group under the operation $\oplus$ defined by $(x_1,x_2--x_m)\oplus(y_1,y_2--y_m)=(x_1+y_1,x_2+y_2,--x_m+y_m)$ observe that B^m has 2^m elements. i.e. order of group B^m is 2^m.

Important Terminology :

Let us choose an integer $n>m$ and one-to-one function $e:B^m\rightarrow B^n$.

1) Encoding Function :

The function e is called an (m, n) encoding function. It means that every word in B^m as a word in B^n.

2) Code word :

If $b\in B^m$ then e(b) is called the code word

3) Weight :

For $x\in B^n$ the number of 1's in x is called the weight of x and is denoted by $|x|$.

e.g. i) $x=10011\in B^5 \therefore w(x)=3$

ii) $x=001\in B^3 \therefore w(x)=1$

4) $x\oplus y\rightarrow$ Let $x,y\in B^n$, then $x\oplus y$ is a sequence of length n that has 1's in those positions x & y differ and has O's in those positions x & y are the same. i.e. The operation + is defined as 0 + 0 = 0   0 + 1 = 1   1 + 1 = 0 1 + 0 = 1 e.g. if $x, y \in B^5$

$x = 00101, y = 10110$

$\therefore x \oplus y = 10011$

$\therefore w(x \oplus y) = 3$

5) Hamming Distance :

Let $x, y \in B^m$. The Hamming Distance $\delta(x, y)$ between x and y is the weight of $x \oplus y$. It is denoted by $|x \oplus y|$. e.g. Hamming distance between x & y can be calculated as follows : if x = 110110, y = 000101 $x \oplus y$ = 110011 so $|x \oplus y|$ = 4.

6) Minimum distance :

Let $x, y \in B^n$.then minimum distance = min $\{d(x, y) / x, y \in B^n\}$. Let $x_1, x_2 -- x_n$ are the code words, let any $x_i, i = 1 --- n$ is a transmitted word and y be the corresponding received word. Then $y = x_k$ if $d(x_k, y)$is the minimum distane for k = 1, 2, --- n. This criteria is known as minimum distance criteria.

7) Detection of errors :

Let $e : B^m \to B^n \ (m < n)$ is an encoding function then if minimum distane of e is (k + 1) then it can detect k or fewer errors.

8) Correction of errors :

Let $e : B^m \to B^n \ (m < n)$ is an encoding function then if minimum distance of e is (2k + 1) then it can correct k or fewer errors.

Weight of a code word : It is the number of 1's present in the given code word.

Hamming distance between two code words : Let $x = x_1 x_2 ... x_m$ and $y = y_1 y_2 ... y_m$ be two code words. The Hamming distance between them, $\delta(x, y)$, is the number of occurrences such that $x_i \neq y_i$ for $i = 1, m$.

Example 7.1 : Find the weights of the following code words.

Example 7.1 : Define weight of a codeword. Find the weights of the following. [Apr-04, May-06]

(a) $x = 010000$ (b) $x = 11100$

(c) $x = 00000$ (d) $x = 11111$

(e) $x = 01001$ (f) $x = 11000$

Solution : Weight of a code word :

(a) $|x| = |010000| = 1$ (b) $|x| = |11100| = 3$

(c) $|x| = |00000| = 0$ (d) $|x| = |11111| = 5$

(e) $|x| = 2$ (f) $|x| = 2$

Example 7.2 : Define Hamming distance. Find the Hamming distance between the codes. [Apr-04]

(a) $x = 010000, \quad y = 000101$ (b) $x = 001100, \quad y = 010110$

Solution : Hamming distance :

(a) $\delta(x, y) = |x \oplus y| = |010000 \oplus 000101| = |010101| = 3$

(b) $\delta(x, y) = |x \oplus y| = |001100 \oplus 010110| = |011010| = 3$

Example 7.3 : Let d be the $(4,3)$ decoding function defined by $d : B^4 \to B^3$. If $y = y_1 y_2 \ldots y_{m+1}$, $d(y) = y_1 y_2 \ldots y_m$.

Determine $d(y)$ for the word y is B^4. [Nov-06]

(a) $\mathrm{y} = 0110$ (b) $\mathrm{y} = 1011$

Solution : (a) $\mathrm{d}(\mathrm{y}) = 011$ (b) $\mathrm{d}(\mathrm{y}) = 101$

Example 7.4 : Let $d : B^6 \to B^2$ be a decoding function defined by for $y = y_1 y_2 \ldots y_6$. Then $d(y) = z_1 z_2$.

where

$\mathrm{zi} = 1$ if $\{y_1, y_{i+2}, y_{i+4}\}$ has at least two 1’s.

0 if $\{y_1, y_{i+2}, y_{i+4}\}$ has less than two 1’s.

Determine $\mathrm{d}(\mathrm{y})$ for the word y in B^6.

(a) $y = 111011$ (b) $y = 010100$

Solution : (a) $d(y) = 11$ (b) $d(y) = 01$

Example 7.5 : The following encoding function $f : B^m \to B^{m+1}$ is called the parity $(m, m+1)$ check code. If $b = b_1 b_2 \ldots b_m \in B^m$, define $e(b) = b_1 b_2 \ldots b_m b_{m+1}$

where

$b_{m+1} = 0$ if $|\mathrm{b}|$ is even.

= 1 if $|\mathrm{b}|$ is odd.

Find $\mathrm{e}(\mathrm{b})$ if (a) $\mathrm{b} = 01010$ (b) $\mathrm{b} = 01110$

Solution : (a) $e(b) = 010100$ (b) $e(b) = 011101$

Example 7.6 : Let $e : B^2 \to B^6$ is an (2,6) encoding function defined as

e(00) = 000000, e(01) = 011101

e(10) = 001110, e(11) = 111111

a) Find minimum distance.

b) How many errors can e detect?

c) How many errors can e correts?

Solution : Let $x_0, x_1, x_2, x_3 \in B^6$ where $x_0 = 000000, x_1 = 011101,$ $x_2 = 001110, x_3 = 111111$

$w(x_0 \oplus x_1) = w(011101) = 4$

$w(x_0 \oplus x_2) = w(001110) = 3$

$w(x_0 \oplus x_3) = w(111111) = 6$

$w(x_1 \oplus x_2) = w(010011) = 3$

$w(x_1 \oplus x_3) = w(100010) = 2$

$w(x_2 \oplus x_3) = w(110001) = 3$

Minimum distance = e = 2

d) Minimum distance = 2

An encoding function e can detect k or fewer errors if the minimum distance is k + 1. $\therefore k + 1 = 2 \therefore k = 1$

$\therefore$ The function can detect 1 or fewer (i.e. 0) error.

e) e can correct k or fewer error if minimum distance is 2k + 1.

$\therefore$ 2k + 1 = 2

$\therefore$ k = $\frac{1}{2}$

$\therefore$ e can correct $\frac{1}{2}$ or less than $\frac{1}{2}$ i.e. 0 errors.

7.2 GROUP CODE :

An (m, n) encoding function $e : B^m \to B^n$ is called a group code if range of e is a subgroup of B^n. i.e. (Ran (e), $\oplus$) is a group.

Since Ran (e) $\subseteq B^n$ and if (Ran (e), $\oplus$) is a group then Ran(e) is a subgroup of B^n. If an encoding function $e : B^m \to B^n$ (n < n) is a group code, then the minimum distance of e is the minimum weight of a nonzero codeword.

7.3 ADDITIONAL RESULTS FROM BOOLEAN MATRICES :

(a) Mod-2 Addition : Consider the set B with +. Now let $D=\left[d_{ij}\right]$ and $E=\left[e_{ij}\right]$ be $m\times n$ Boolean matrices. We denote the mod-2 sum $\mathrm{D}\oplus\mathrm{E}$ as the $\mathrm{m}\times\mathrm{n}$ Boolean matrix $F=\left[f_{ij}\right]$.

where

$$f_{ij}=d_{ij}+e_{ij}, \qquad 1\le i\le m, \qquad 1\le j\le n$$

Here + is addition in B.

For example

$$\begin{bmatrix}1&0&1&0\\0&1&1&0\\1&0&0&1\end{bmatrix}\oplus\begin{bmatrix}1&1&0&1\\1&1&0&1\\0&1&1&1\end{bmatrix}=\begin{bmatrix}0&1&1&0\\0&1&1&1\\1&1&1&0\end{bmatrix}$$

(b) Mod-2 Product : $\mathrm{D}*\mathrm{E}$

$$\begin{bmatrix}1&1&0\\0&1&1\end{bmatrix}*\begin{bmatrix}1&0\\1&1\\0&1\end{bmatrix}=\begin{bmatrix}0&1\\1&0\end{bmatrix}$$

Theorem : Let D and E be $\mathrm{m}\times\mathrm{p}$ Boolean matrices, and F be a $\mathrm{p}\times\mathrm{n}$ Boolean matrix. Then

$$(D\oplus E)*F=(D*F)\oplus(E*F)$$

That is distributive property holds for $\oplus$ and $*$.

Theorem 7.1 : Let m and n be non-negative integers with $m<n$, $r=n-m$ and let H be an $n\times r$ Boolean matrix. Then the function $f_H : B^n\to B^r$ defined by $f_H(x)=x*H, \quad x\in B^n$ is a homomorphism from the group B^{n} to the group $\mathrm{B}^{'}$.

Proof : Let x and y be elements in B^{n} then

$$\begin{aligned}f_H(x)&=(x\oplus y)*H\\&=(x*H)\oplus(y*H)\\&=f_H(x)\oplus f_H(y)\end{aligned}$$

Hence, f_{H} is a homomorphism from the group B^{n} to the group $\mathrm{B}^{'}$.

Corollary 7.1 : Let m, n, r, H and $\mathrm{f_h}$ be as in Theorem 2. Then $N = \left\{ x \in B^n / x * H = 0 \right\}$ is a normal subgroup of $\mathrm{B^n}$.

Parity Check Matrix : Let $\mathrm{m < n}$ and $\mathrm{r = n - m}$. An $\mathrm{n \times r}$ Boolean matrix

$$\mathrm{H} = \begin{bmatrix} \mathrm{h_{11}} & \mathrm{h_{12}} & . & . & . & \mathrm{h_{1r}} \\ \mathrm{h_{21}} & \mathrm{h_{22}} & . & . & . & \mathrm{h_{2r}} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \mathrm{h_{m1}} & \mathrm{h_{m2}} & . & . & . & \mathrm{h_{mr}} \\ 1 & 0 & . & . & . & 0 \\ 0 & 1 & . & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & . & . & . & 0 \end{bmatrix}$$

whose last r rows form $\mathrm{r \times r}$ identity matrix is called a parity check matrix.

we use H to define an encoding function $e_H : B^m \to B^n$. If $b = b_1 b_2 \ldots b_m$, let $x = e_H(b) = b_1 b_2 \ldots b_m x_1 x_2 \ldots x_r$,

where

$$x_1 = b_1 h_{11} + b_2 h_{21} + \ldots + b_m h_{m1}$$
$$x_2 = b_1 h_{12} + b_2 h_{22} + \ldots + b_m h_{m2}$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$x_r = b_1 h_{1r} + b_2 h_{2r} + \ldots + b_m h_{mr}$$

Theorem 7.2 : Let $x = y_1 y_2 \ldots y_m b_m x_1 x_2 \ldots x_1 \in B^n$. Then $x * H = 0$ if and only $x = e_H(b)$ for some $b \in B^m$.

Corollary 7.2 : $e_H\left(B^m\right) = \left\{ e_H(b) / b \in B^m \right\}$ is a subgroup of $\mathrm{B^n}$.

7.4 DECODING AND ERROR CORRECTION :

Consider an $\left(\mathrm{m}, \mathrm{n}\right)$ encoding function $e : B^m \to B^n$, we require an (n,m) decoding function associate with e as $d : B^n \to B^m$.

The method to determine a decoding function d is called maximum likelihood technique.

Since $\left|\mathrm{B}^{\mathrm{m}}\right| = 2^{\mathrm{m}}$.

Let $\mathrm{x_k} \in \mathrm{B}^{\mathrm{m}}$ be a codeword, k = 1, 2, ---$^{\mathrm{m}}$ and the received word is y then. Min $1 \le \mathrm{k} \le 2^{\mathrm{m}} \left\{\mathrm{d}\left(\mathrm{x_k}, \mathrm{y}\right)\right\} = \mathrm{d}\left(\mathrm{x_i}, \mathrm{y}\right)$ for same i then x_i is a codeword which is closest to y. If minimum distance is not unique then select on priority

7.5 MAXIMUM LIKELIHOOD TECHNIQUE :

Given an $\left(\mathrm{m}, \mathrm{n}\right)$ encoding function $e : B^m \to B^n$, we often need to determine an $\left(\mathrm{n}, \mathrm{m}\right)$ decoding function $d : B^n \to B^m$ associated with e. We now discuss a method, called the maximum likelihood techniques, for determining a decoding function d for a given e. Since B^{m} has 2^{m} elements, there are 2^{m} code words in B^{n}. We first list the code words in a fixed order.

$$x^{(1)}, x^{(2)}, \ldots, x^{\left(2^m\right)}$$

If the received word is x_1, we compute $\delta\left(x^{(i)}, x_1\right)$ for $1 \le i \le 2^m$ and choose the first code word, say it is $x^{(s)}$, such that

$$\min_{1 \le i \le 2^m} \left\{\delta\left(x^{(i)}, x_1\right)\right\} = \delta\left(x^{(s)}, x_1\right)$$

That is, $x^{(s)}$ is a code word that is closest to x_1, and the first in the list. If $x^{(s)} = e\left(b\right)$, we define the maximum likelihood decoding function d associated with e by

$$d\left(x_t\right) = b$$

Observe that d depends on the particular order in which the code words in $e\left(B^n\right)$ are listed. If the code words are listed in a different order, we may obtain, a different likelihood decoding function d associated with e.

Theorem 7.3 : Suppose that e is an (m,n) encoding function and d is a maximum likelihood decoding function associated with e. Then (e,d) can correct k or fewer errors if and only if the minimum distance of e is at least $2\mathrm{k}+1$.

Example 7.7 : Let $\mathrm{m}=2, \mathrm{n}=5$ and $\mathrm{H}=\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Determine the group code $\mathrm{e_H} : \mathrm{B}^2 \to \mathrm{B}^5$. [May-07]

Solution : We have $\mathrm{B}^2 = \{00, 01, 10, 11\}$. Then $\mathrm{e}(00) = 00\mathrm{x_1 x_2 x_3}$ where

$$x_1 = 0.1 + 0.0 = 0$$
$$x_2 = 0.1 + 0.1 = 0$$
$$x_3 = 0.0 + 0.1 = 0$$
$$\therefore\ e(00) = 00000$$

Now,

$$e(01) = 01x_1x_2x_3$$

where

$$x_1 = 0.1 + 1.0 = 0$$
$$x_2 = 0.1 + 1.1 = 1$$
$$x_3 = 0.0 + 1.1 = 1$$
$$\therefore\ e(01) = 01011$$

Next

$$e(10) = 10x_1x_2x_3$$
$$x_1 = 1.1 + 0.0 = 1$$
$$x_2 = 1.1 + 1.0 = 1$$
$$x_3 = 1.0 + 0.1 = 0$$
$$\therefore\ e(10) = 10110$$
$$e(11) = 11101$$

Example 7.8 : Let $H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ be a parity check matrix. determine the $(3, 6)$ group code $e_H : B^3 \rightarrow B^6$.

Solution : First find $e(000)$, $e(001)$, $e(010)$, $e(011)$, $e(100)$, $e(101)$, $e(110)$, $e(111)$.

$e(000) = 000000$	$e(100) = 100100$
$e(001) = 001111$	$e(101) = 101011$
$e(010) = 010011$	$e(110) = 110111$
$e(100) = 011100$	$e(111) = 111000$

Example 7.9 : Consider the group code defined by $e : B^2 \rightarrow B^5$ such that $e(00) = 00000$ $e(01) = 01110$ $e(10) = 10101$ $e(11) = 11011$. Decode the following words relative to maximum likelihood decoding function.

(a) 11110 (b) 10011 (c) 10100

Solution : (a) $x_t = 1110$

Compute $\delta\left(x^{(1)}, x_t\right) = |\,00000 \oplus 11110\,| = |\,11110\,| = 4$

$$\delta\left(x^{(2)}, x_t\right) = |\,01110 \oplus 11110\,| = |\,10000\,| = 1$$

$$\delta\left(x^{(3)}, x_t\right) = |\,10101 \oplus 11110\,| = |\,01011\,| = 3$$

$$\delta\left(x^{(4)}, x_t\right) = |\,11011 \oplus 11110\,| = |\,00101\,| = 2$$

$$\min\left\{\delta\left(x^{(i)}, x_t\right)\right\} = 1 = \delta\left(x^{(2)}, x_t\right)$$

$\therefore$ $e(01) = 01110$ is the code word closest to $x_t = 11110$.

$\therefore$ The maximum likelihood decoding function d associated with e is defined by $d(x_t) = 01$.

(b) $x_t = 10011$

Compute $\delta\left(x^{(1)}, x_t\right) = |\, 00000 \oplus 10011 \,| = |\, 11101 \,| = 4$

$$\delta\left(x^{(2)}, x_t\right) = |\, 01110 \oplus 10011 \,| = |\, 00110 \,| = 2$$

$$\delta\left(x^{(3)}, x_t\right) = |\, 10101 \oplus 11110 \,| = |\, 01011 \,| = 3$$

$$\delta\left(x^{(4)}, x_t\right) = |\, 11011 \oplus 10011 \,| = |\, 01000 \,| = 1$$

$$\min\left\{\delta\left(x^{(i)}, x_t\right)\right\} = 1 = \delta\left(x^{(4)}, x_t\right)$$

$\therefore$ $e(11) = 11011$ is the code word closest to $x_t = 10011$.

$\therefore$ The maximum likelihood decoding function d associated with e is defined by $d\left(x_t\right) = 11$.

(c) $x_t = 10100$

Compute $\delta\left(x^{(1)}, x_t\right) = |\, 00000 \oplus 10100 \,| = |\, 10100 \,| = 2$

$$\delta\left(x^{(2)}, x_t\right) = |\, 01110 \oplus 10100 \,| = |\, 11010 \,| = 3$$

$$\delta\left(x^{(3)}, x_t\right) = |\, 10101 \oplus 10100 \,| = |\, 00001 \,| = 1$$

$$\delta\left(x^{(4)}, x_t\right) = |\, 11011 \oplus 10100 \,| = |\, 01111 \,| = 4$$

$$\min\left\{\delta\left(x^{(i)}, x_t\right)\right\} = 1 = \delta\left(x^{(3)}, x_t\right)$$

$\therefore$ $e(10) = 10101$ is the code word closest to $x_t = 10100$.

$\therefore$ The maximum likelihood decoding function d associated with e is defined by $d\left(x_t\right) = 10$.

Example 7.10 : Let $H = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ be a parity check matrix. decode the following words relative to a maximum likelihood decoding function associated with e_H : (i) 10100, (ii) 01101, (iii) 11011.

Solution : The code words are $e(00) = 00000$, $e(01) = 00101$, $e(10) = 10011$, $e(11) = 11110$. Then $N = \{00000, 00101, 10011, 11110\}$. We implement the decoding procedure as follows. Determine all left cosets of N in B5, as rows of a table. For each row 1, locate the coset leader ε_i, and rewrite the row in the order.

$\varepsilon_1, \varepsilon_i \oplus$

Example 7.11 : Consider the $(2,4)$ encoding function e as follows. How many errors will e detect? [May-06]

$e(00) = 0000,\ e(01) = 0110,\ e(10) = 1011,\ e(11) = 1100$

Solution :

$\oplus$	0000	0110	1011	1100
0000	---	0110	1011	1100
0110		---	1101	1010
1011			---	0111
1100				---

Minimum distance between distinct pairs of $e = 2 \quad \therefore k+1 = 2 \quad \therefore k = 1$.
$\therefore$ the encoding function e can detect 1 or fewer errors.

Example 7.12 : Define group code. Show that $(2,5)$ encoding function $e : B^2 \to B^5$ defined by $e(00) = 0000,\ e(10) = 10101,\ e(11) = 11011$ is a group code. [May-06]

Solution : Group Code

$\oplus$	00000	01110	10101	11011
00000	00000	01110	10101	11011
01110	01110	00000	11011	10101
10101	10101	11011	00000	01110
11011	11011	10101	01110	00000

Since closure property is satisfied, it is a group code.

Example 7.13 : Define group code. show that $(2,5)$ encoding function $e : B^2 \to B^5$ defined by $e(00) = 00000,\ e(01) = 01110,\ e(10) = 10101,$

$\mathrm{e}(11)=11011$ is a group code. Consider this group code and decode the following words relative to maximum likelihood decoding function.
(a) 11110 (b) 10011. [Apr-04]

Solution : Group Code

$\oplus$	00000	01110	10101	11011
00000	00000	01110	10101	11011
01110	01110	00000	11011	10101
10101	10101	11011	00000	01110
11011	11011	10101	01110	00000

Since closure property is satisfied, it is a group code.

Now, let $x^{(1)}=00000,\ x^{(2)}=01110,\ x^{(3)}=10101,\ x^{(4)}=11011$.

(a) $x_t=11110$

$$\delta\left(x^{(1)},x_t\right)=\left|x^{(1)}\oplus x_t\right|=|00000\oplus 11110|=|11110|=4$$
$$\delta\left(x^{(2)},x_t\right)=\left|x^{(2)}\oplus x_t\right|=|01110\oplus 1110|=|10000|=1$$
$$\delta\left(x^{(3)},x_t\right)=\left|x^{(3)}\oplus x_t\right|=|10101\oplus 1110|=|01011|=3$$
$$\delta\left(x^{(4)},x_t\right)=\left|x^{(4)}\oplus x_t\right|=|11011\oplus 1110|=|00101|=2$$

$\therefore$ Maximum likelihood decoding function $\mathrm{d}(\mathrm{x_t})=01$.

(b) $x_t=10011$

$$\delta\left(x^{(1)},x_t\right)=\left|x^{(1)}\oplus x_t\right|=|00000\oplus 10011|=|10011|=3$$
$$\delta\left(x^{(2)},x_t\right)=\left|x^{(2)}\oplus x_t\right|=|01110\oplus 10011|=|11101|=4$$
$$\delta\left(x^{(3)},x_t\right)=\left|x^{(3)}\oplus x_t\right|=|10101\oplus 10011|=|00110|=2$$
$$\delta\left(x^{(4)},x_t\right)=\left|x^{(4)}\oplus x_t\right|=|11011\oplus 10011|=|01000|=1$$

$\therefore$ Maximum likelihood decoding function $d(x_t)=11$.

Example 7.14 : Let $H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ be a parity check matrix. Determine the $(3, 6)$ group code $e_H : B^3 \to B^6$.

Solution : $B^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

$e_H(000) = 000000$ $\quad$ $e_H(001) = 001111$ $\quad$ $e_H(010) = 010011$

$e_H(011) = 011100$ $\quad$ $e_H(100) = 100100$ $\quad$ $e_H(101) = 101011$

$e_H(110) = 110111$ $\quad$ $e_H(111) = 111000$

$\therefore$ Required group code = $\{000000, 001111, 010011, 011100, 100100, 101011, 110111, 111000\}$

Example 7.15 : Show that $(2, 5)$ encoding function $e : B_2 \to B_5$ defined by $e(00) = 00000$, $e(01) = 01110$, $e(10) = 10101$, $e(11) = 11011$ is a group code. [May-06]

OR

Test whether the following $(2, 5)$ encoding function is a group code.

$e(00) = 00000$, $e(01) = 01110$, $e(10) = 10101$, $e(11) = 11011$ [Oct-03]

Solution :

$\oplus$	00000	01110	10101	11011
00000	00000	01110	10101	11011
01110	01110	00000	11011	10101
10101	10101	11011	00000	01110
11011	11011	10101	01110	00000

Since closure property is satisfied, it is a group code.

Example 7.16 : Show that the $(3, 7)$ encoding function $e : B^3 \to B^7$ defined by

$e(000) = 0000000$ $\quad$ $e(001) = 0010110$ $\quad$ $e(010) = 0101000$

$e(011) = 0111110$ $e(100) = 1000101$ $e(101) = 1010011$
$e(110) = 1101101$ $e(111) = 1111011$ is a group code.

Solution :

$\oplus$	0000000	0010110	0101000	0111110	1000101	1010011	1101101	1111011
0000000	0000000	0010110	0101000	0111110	1000101	1010011	1101101	1111011
0010110	0010110	0000000	0111110	0101000	1010011	1000101	1111011	1101101
0101000	0101000	0111110	0000000	0010110	1101101	1111011	1000101	1010011
0111110	0111110	0101000	0010110	0000000	1111011	1101101	1010011	1000101
1000101	1000101	1010011	1101101	1111011	0000000	0010110	0101000	0111110
1010011	1010011	1000101	1111011	1101101	0010110	0000000	0111110	0101100
1101101	1101101	1111011	1000101	1010011	0101000	0111110	0000000	0010110
1111011	1111011							0000000

Since closure property is satisfied, it is a group code.

Example 7.17 : Consider the $(3,8)$ encoding function $e : B^3 \to B^8$ defined by

$e(000) = 0000000$ $e(100) = 10100100$ $e(001) = 10111000$
$e(101) = 10001001$ $e(010) = 00101101$ $e(110) = 00011100$
$e(011) = 10010101$ $e(111) = 00110001$.

How many errors will e detect?

Solution :

$\oplus$	00000000	10100100	10111000	10001001	00101101	00011100	10010101	00110001
0000000	00000000	10100100	10111000	10001001	00101101	00011100	10010101	00110001
10100100	10100100	00000000	00011100	00101101	10001001	10111000	00110001	10010101
10111000	00000000	00011100	00000000	001100001	10010101	10100100	00101101	10001001
10001001	10001001	00101101	00110001	00000000	10100100	10010101	00011100	10111000
00101101	00101101	10001001	10010101	10100100	00000000	00110001	10111000	00011100
00011100	00011100	10111000	10100100	10010101	00110001	00000000	10001001	00101101
10010101	10010101	00110001	00101101	00011100	10111000	10001001	00000000	10100100
00110001	00110001	10010101	10001001	10111000	00011100	00101101	10100100	0000000

Minimum distance between pairs of $e = 3$.
$\therefore k+1=3$ $\therefore k=2$ $\therefore$ The encoding function e can detect 2 or fewer errors.

Example 7.18 : Consider parity check matrix H given by $H = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Determine the group code $e_H : B_2 \to B_5$. Decode the following words relative to a maximum likelihood decoding function associated with $e_H : 01110,\ 11101,\ 00001,\ 11000$. [Apr-04, May-07]

Solution : $B_2 = \{00, 01, 10, 11\}$

$e_H(00) = 00x_1x_2x_3$ where $x_1 = 0.1 + 0.0 = 0$
$x_2 = 0.1 + 0.1 = 0$
$x_3 = 0.0 + 0.1 = 0$ $\quad \therefore e_H(00) = 00000$

$e_H(01) = 01x_1x_2x_3$ where $x_1 = 0.1 + 1.0 = 0$
$x_2 = 0.1 + 1.1 = 1$
$x_3 = 0.0 + 1.1 = 1$ $\quad \therefore e_H(01) = 01011$

$e_H(10) = 10x_1x_2x_3$ where $x_1 = 1.1 + 0.0 = 1$
$x_2 = 1.1 + 0.1 = 1$
$x_3 = 1.0 + 0.1 = 0$ $\quad \therefore e_H(01) = 10110$

$e_H(11) = 11x_1x_2x_3$ where $x_1 = 1.1 + 1.0 = 1$
$x_2 = 1.1 + 1.1 = 0$
$x_3 = 1.0 + 1.1 = 1$ $\quad \therefore e_H(01) = 11101$

$\therefore$ Desired group code = $\{00000,\ 01011,\ 10110,\ 11101\}$

(1) $x_t = 01110$

$\delta\left(x^{(1)}, x_t\right) = \left| x^{(1)} \oplus x_t \right| = |\, 00000 \oplus 01110 \,| = |\, 01110 \,| = 3$

$\delta\left(x^{(2)}, x_t\right) = \left| x^{(2)} \oplus x_t \right| = |\, 01011 \oplus 01110 \,| = |\, 00101 \,| = 2$

$\delta\left(x^{(3)}, x_t\right) = \left| x^{(3)} \oplus x_t \right| = |\, 10110 \oplus 01110 \,| = |\, 11000 \,| = 2$

$\delta\left(x^{(4)}, x_t\right) = \left| x^{(4)} \oplus x_t \right| = |\, 11101 \oplus 01110 \,| = |\, 10011 \,| = 3$

$\therefore$ Maximum likelihood decoding function $d(x_t) = 01$

(2) $x_t = 11101$

$$\delta\left(x^{(1)}, x_t\right) = \left| x^{(1)} \oplus x_t \right| = |\,00000 \oplus 11101\,| = |\,11101\,| = 4$$

$$\delta\left(x^{(2)}, x_t\right) = \left| x^{(2)} \oplus x_t \right| = |\,01110 \oplus 11101\,| = |\,10110\,| = 3$$

$$\delta\left(x^{(3)}, x_t\right) = \left| x^{(3)} \oplus x_t \right| = |\,10101 \oplus 11101\,| = |\,01011\,| = 3$$

$$\delta\left(x^{(4)}, x_t\right) = \left| x^{(4)} \oplus x_t \right| = |\,11011 \oplus 11101\,| = |\,00000\,| = 0$$

$\therefore$ Maximum likelihood decoding function $\mathrm{d}\left(\mathrm{x}_t\right) = 11$

(3) $x_t = 00001$

$$\delta\left(x^{(1)}, x_t\right) = \left| x^{(1)} \oplus x_t \right| = |\,00000 \oplus 00001\,| = |\,00001\,| = 1$$

$$\delta\left(x^{(2)}, x_t\right) = \left| x^{(2)} \oplus x_t \right| = |\,01011 \oplus 00001\,| = |\,01010\,| = 2$$

$$\delta\left(x^{(3)}, x_t\right) = \left| x^{(3)} \oplus x_t \right| = |\,10110 \oplus 00001\,| = |\,10111\,| = 4$$

$$\delta\left(x^{(4)}, x_t\right) = \left| x^{(4)} \oplus x_t \right| = |\,11101 \oplus 00001\,| = |\,11100\,| = 3$$

$\therefore$ Maximum likelihood decoding function $\mathrm{d}\left(\mathrm{x}_t\right) = 00$

(2) $x_t = 11000$

$$\delta\left(x^{(1)}, x_t\right) = \left| x^{(1)} \oplus x_t \right| = |\,00000 \oplus 11000\,| = |\,11000\,| = 2$$

$$\delta\left(x^{(2)}, x_t\right) = \left| x^{(2)} \oplus x_t \right| = |\,01110 \oplus 11000\,| = |\,10011\,| = 3$$

$$\delta\left(x^{(3)}, x_t\right) = \left| x^{(3)} \oplus x_t \right| = |\,10101 \oplus 11000\,| = |\,01101\,| = 3$$

$$\delta\left(x^{(4)}, x_t\right) = \left| x^{(4)} \oplus x_t \right| = |\,11011 \oplus 11000\,| = |\,10000\,| = 1$$

$\therefore$ Maximum likelihood decoding function $d\left(x_t\right) = 11$

Example 7.19 : Let $\mathrm{H} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$ be a parity check matrix. decode 0110 relative to a maximum likelihood decoding function associated with $\mathrm{e_H}$.

[Dec-04]

Solution : $e_H : B_2 \to B_5$

$B_2 = \{00, 01, 10, 11\}$

$e_H(00) = 00x_1x_2$ where $x_1 = 0.1 + 0.0 = 0$

$x_2 = 0.1 + 0.1 = 0$ $\therefore e_H(00) = 0000$

$e_H(01) = 01x_1x_2$ where $x_1 = 0.1 + 1.0 = 0$

$x_2 = 0.1 + 1.1 = 1$ $\therefore e_H(01) = 0101$

$e_H(10) = 10x_1x_2$ where $x_1 = 1.1 + 0.0 = 1$

$x_2 = 1.1 + 0.1 = 1$ $\therefore e_H(01) = 1011$

$e_H(11) = 11x_1x_2$ where $x_1 = 1.1 + 1.0 = 1$

$x_2 = 1.1 + 1.1 = 0$ $\therefore e_H(01) = 1110$

Let $x^{(1)} = 0000,\ x^{(2)} = 0101,\ x^{(3)} = 1011,\ x^{(4)} = 1110$.
Let $x_1 = 0110$.

$$\delta\left(x^{(1)}, x_t\right) = \left| x^{(1)} \oplus x_t \right| = |\,0000 \oplus 0110\,| = |\,0110\,| = 2$$
$$\delta\left(x^{(2)}, x_t\right) = \left| x^{(2)} \oplus x_t \right| = |\,0101 \oplus 0110\,| = |\,0011\,| = 2$$
$$\delta\left(x^{(3)}, x_t\right) = \left| x^{(3)} \oplus x_t \right| = |\,1011 \oplus 0110\,| = |\,1011\,| = 3$$
$$\delta\left(x^{(4)}, x_t\right) = \left| x^{(4)} \oplus x_t \right| = |\,1110 \oplus 0110\,| = |\,1000\,| = 1$$
$$\therefore Min\,\delta\left(x^{(i)}, x_t\right) = \delta\left(x^{(4)}, x_t\right) \ and \ e(11) = x^{(4)} \qquad \therefore d(x_t) = 11.$$

Example 7.20 : Consider the $(2, 5)$ group encoding function defined by $\mathrm{e}(00) = 00000,\ \mathrm{e}(01) = 01101,\ \mathrm{e}(10) = 10011,\ \mathrm{e}(11) = 11110$ and d be an associated maximum likelihood function. Use d to decode the following words. [May-03, May-05]

(i) 10100 (ii) 01101

Solution : Let $x^{(1)} = 00000,\ x^{(2)} = 01011,\ x^{(3)} = 10110,\ x^{(3)} = 11110$

(1) $x_t = 10100$

$$\delta\left(x^{(1)}, x_t\right) = \left| x^{(1)} \oplus x_t \right| = |\,00000 \oplus 10100\,| = |\,10100\,| = 2$$
$$\delta\left(x^{(2)}, x_t\right) = \left| x^{(2)} \oplus x_t \right| = |\,01101 \oplus 10100\,| = |\,11001\,| = 3$$

$\delta\left(x^{(3)}, x_t\right) = \left| x^{(3)} \oplus x_t \right| = |10011 \oplus 10100| = |00111| = 3$

$\delta\left(x^{(4)}, x_t\right) = \left| x^{(4)} \oplus x_t \right| = |11110 \oplus 10100| = |01010| = 2$

$\therefore\ Min\,\delta\left(x^{(i)}, x_t\right) = \delta\left(x^{(1)}, x_t\right)$ i.e. $x^{(1)}$ is the code word which is closest to x_t and $1 \le \mathrm{i} \le 4$

The first in their list in the list and $e(00) = x^{(1)}$. So we define maximum likelihood decoding function d associated with e by $\mathrm{d}(\mathrm{x}_\mathrm{t}) = 00$.

(2) $x_t = 01100$

$\delta\left(x^{(1)}, x_t\right) = \left| x^{(1)} \oplus x_t \right| = |00000 \oplus 01101| = |01101| = 3$

$\delta\left(x^{(2)}, x_t\right) = \left| x^{(2)} \oplus x_t \right| = |01101 \oplus 01101| = |00000| = 0$

$\delta\left(x^{(3)}, x_t\right) = \left| x^{(3)} \oplus x_t \right| = |10011 \oplus 01101| = |11110| = 4$

$\delta\left(x^{(4)}, x_t\right) = \left| x^{(4)} \oplus x_t \right| = |11110 \oplus 01101| = |10011| = 3$

$\therefore\ \mathrm{Min}\,\delta\left(\mathrm{x}^{(\mathrm{i})}, \mathrm{x}_\mathrm{t}\right) = \delta\left(\mathrm{x}^{(2)}, \mathrm{x}_\mathrm{t}\right)$ i.e. $x^{(2)}$ is the code word which is closest to x_t and $1 \le \mathrm{i} \le 4$

The first in their list in the list and $e(01) = x^{(2)}$. So we define maximum likelihood decoding function d associated with e by $d(x_t) = 01$.

Example 7.21 : Let $\mathrm{H} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$ be a parity check matrix. [Dec-02]

i) Determine the $(3, 5)$ group code $\mathrm{e_H} : \mathrm{B}^3 \to \mathrm{B}^5$.

ii) Construct the decoding table and decode the following words using maximum likelihood technique – 1) 00111, 2) 10111, 3) 11001

Solution : (i) $\mathrm{e_H} : \mathrm{B}^3 \to \mathrm{B}^5$.

$\mathrm{B}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

$e_H(000) = 000x_1x_2$ where $x_1 = 0.1 + 0.0 + 0.1 = 0$

$x_2 = 0.1 + 0.1 + 0.0 = 0 \ \therefore \mathrm{e_H}(000) = 00000$

$e_H(001) = 001x_1x_2$ where $x_1 = 0.1 + 0.0 + 1.1 = 1$

$x_2 = 0.1 + 0.1 + 1.0 = 0 \ \therefore \mathrm{e_H}(001) = 00110$

$e_H(010) = 010x_1x_2$ where $x_1 = 0.1 + 1.0 + 0.1 = 0$

$x_2 = 0.1 + 1.1 + 0.0 = 1 \ \therefore \mathrm{e_H}(010) = 01001$

$e_H(011) = 011x_1x_2$ where $x_1 = 0.1 + 1.0 + 1.1 = 1$

$x_2 = 0.1 + 1.1 + 1.0 = 1 \ \therefore \mathrm{e_H}(011) = 01111$

$e_H(100) = 100x_1x_2$ where $x_1 = 1.1 + 0.0 + 0.1 = 1$

$x_2 = 1.1 + 0.1 + 0.0 = 1 \ \therefore \mathrm{e_H}(100) = 10011$

$e_H(101) = 101x_1x_2$ where $x_1 = 1.1 + 0.0 + 1.1 = 0$

$x_2 = 1.1 + 0.1 + 1.0 = 1 \ \therefore \mathrm{e_H}(001) = 10101$

$e_H(110) = 110x_1x_2$ where $x_1 = 1.1 + 1.0 + 0.1 = 1$

$x_2 = 1.1 + 1.1 + 1.0 = 0 \ \therefore \mathrm{e_H}(110) = 11010$

$e_H(111) = 111x_1x_2$ where $x_1 = 1.1 + 1.0 + 1.1 = 0$

$x_2 = 1.1 + 1.1 + 1.0 = 0 \ \therefore \mathrm{e_H}(111) = 11100$

Let $x^{(1)} = 00000,\ x^{(2)} = 00110,\ x^{(3)} = 01001,\ x^{(4)} = 01111$

$x^{(5)} = 10011,\ x^{(6)} = 10101,\ x^{(7)} = 11010,\ x^{(8)} = 11100$

(ii) (1) Let $x_t = 00111$

$$\delta\left(x^{(1)}, x_t\right) = \left| x^{(1)} \oplus x_t \right| = |\,00111\,| = 3$$

$$\delta\left(x^{(2)}, x_t\right) = \left| x^{(2)} \oplus x_t \right| = |\,00001\,| = 1$$

$$\delta\left(x^{(3)}, x_t\right) = \left| x^{(3)} \oplus x_t \right| = |\,01110\,| = 3$$

$$\delta\left(x^{(4)}, x_t\right) = \left| x^{(4)} \oplus x_t \right| = |\, 01000 \,| = 1$$

$$\delta\left(x^{(5)}, x_t\right) = \left| x^{(5)} \oplus x_t \right| = |\, 10100 \,| = 2$$

$$\delta\left(x^{(6)}, x_t\right) = \left| x^{(6)} \oplus x_t \right| = |\, 10010 \,| = 2$$

$$\delta\left(x^{(7)}, x_t\right) = \left| x^{(7)} \oplus x_t \right| = |\, 11101 \,| = 4$$

$$\delta\left(x^{(8)}, x_t\right) = \left| x^{(8)} \oplus x_t \right| = |\, 11011 \,| = 4$$

(2) Let $x_t = 10111$

$$\delta\left(x^{(1)}, x_t\right) = \left| x^{(1)} \oplus x_t \right| = |\, 10111 \,| = 4$$

$$\delta\left(x^{(2)}, x_t\right) = \left| x^{(2)} \oplus x_t \right| = |\, 10001 \,| = 2$$

$$\delta\left(x^{(3)}, x_t\right) = \left| x^{(3)} \oplus x_t \right| = |\, 11110 \,| = 4$$

$$\delta\left(x^{(4)}, x_t\right) = \left| x^{(4)} \oplus x_t \right| = |\, 11000 \,| = 2$$

$$\delta\left(x^{(5)}, x_t\right) = \left| x^{(5)} \oplus x_t \right| = |\, 00100 \,| = 1$$

$$\delta\left(x^{(6)}, x_t\right) = \left| x^{(6)} \oplus x_t \right| = |\, 00010 \,| = 1$$

$$\delta\left(x^{(7)}, x_t\right) = \left| x^{(7)} \oplus x_t \right| = |\, 01101 \,| = 3$$

$$\delta\left(x^{(8)}, x_t\right) = \left| x^{(8)} \oplus x_t \right| = |\, 01011 \,| = 3$$

(3) Let $x_t = 11001$

$$\delta\left(x^{(1)}, x_t\right) = \left| x^{(1)} \oplus x_t \right| = |\, 11001 \,| = 3$$

$$\delta\left(x^{(2)}, x_t\right) = \left| x^{(2)} \oplus x_t \right| = |\, 11111 \,| = 5$$

$$\delta\left(x^{(3)}, x_t\right) = \left| x^{(3)} \oplus x_t \right| = |\, 10000 \,| = 1$$

$$\delta\left(x^{(4)}, x_t\right) = \left| x^{(4)} \oplus x_t \right| = |\, 10110 \,| = 3$$

$$\delta\left(x^{(5)}, x_t\right) = \left| x^{(5)} \oplus x_t \right| = |\, 01010 \,| = 2$$

$$\delta\left(x^{(6)}, x_t\right) = \left| x^{(6)} \oplus x_t \right| = |\, 01100 \,| = 2$$

$$\delta\left(x^{(7)}, x_t\right) = \left| x^{(7)} \oplus x_t \right| = |\, 00011 \,| = 2$$

$\delta\left(x^{(8)}, x_t\right) = \left| x^{(8)} \oplus x_t \right| = |\,00101\,| = 2$

$\therefore Min\,\delta\left(x^{(i)}, x_t\right) = \delta\left(x^{(3)}, x_t\right)$ *and* $e(010) = x^{(3)}$ $\quad\therefore d(x_t) = 010$.

Example 7.22 : Let $H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ be a parity check matrix. determine the corresponding group code.

i) How many errors will the above group code detect?
ii) Explain the decoding procedure with an example. [Oct-03]

Solution : Given H is a parity check matrix of $(3, 6)$ group code. $e_H : B^3 \to B^6$.

$B^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$
$e_H(000) = 000000,\ e_H(001) = 001011,\ e_H(010) = 010101,\ e_H(011) = 011111$
$e_H(100) = 100110,\ e_H(101) = 101110,\ e_H(110) = 110011,\ e_H(111) = 111000$.

(i) Min distance of a group code = min weight of non-zero code word = 3
$\therefore k + 1 = 3 \qquad \therefore k = 2$
$\therefore$ The group code can detect at the most 2 or fewer errors.

(ii) Maximum likelihood decoding procedure :
Let $e_H(000) = x^{(1)},\ e_H(001) = x^{(2)},\ e_H(010) = x^{(3)},\ e_H(011) = x^{(4)}$
$e_H(100) = x^{(5)},\ e_H(101) = x^{(6)},\ e_H(110) = x^{(7)},\ e_H(111) = x^{(8)}$
and let x_t be transmitted codeword. Find $\delta\left(x^{(i)}, x_t\right)$, take minimum.

If $Min\,\delta\left(x^{(i)}, x_t\right) = \delta\left(x^{(s)}, x_t\right)$ then maximum likelihood decoding function d can be defined as $d(x_t) = b$ where $e_H(b) = x^{(s)}$. If two or more $x^{(i)}$ have the same minimum value then we select the $x^{(s)}$ whichever comes first in the list and define the decoding function accordingly.

Example 7.23 : Consider $(3,6)$ encoding function e as follows. [May-07]

$e(000)=000000,\ \ e(001)=000110,\ \ e(010)=010010,\ \ e(011)=010100$
$e(100)=100101,\ \ e(101)=100011,\ \ e(110)=110111,\ \ e(111)=110001$

i) Show that the encoding function e is a group code.

ii) Decode the following words with maximum likelihood technique : 101101, 011011.

Solution : (i)

$\oplus$	000000	000110	010010	010100	100101	100011	110111	110001
000000	000000	000110	010010	010100	100101	100011	110111	110001
000110	000110	000000	010100	010010	100011	100101	110001	110111
010010	010010	010100	000000	000110	110111	110001	100101	100011
010100	010100	010010	000110	000000	110001	110111	100011	100101
100101	100101	100011	110111	110001	000000	000110	010010	010100
100011	100011	100101	110001	110111	000110	000000	010100	010010
110111	110111	110001	100101	100011	010010	010100	000000	000110
110001	110001	110111	100011	100101	010100	010010	000110	000000

(ii)

Let $x^{(1)}=000000,\ x^{(2)}=000110,\ x^{(3)}=010010,\ \ x^{(4)}=010100,$ $x^{(5)}=100101,\ x^{(6)}=100011,\ x^{(7)}=110111,\ x^{(8)}=110001.$

(1) Let $x_1=101101$

$$\delta\left(x^{(1)},x_1\right)=\left|x^{(1)}\oplus x_1\right|=|000000\oplus 101101|=|101101|=4$$

$$\delta\left(x^{(2)},x_1\right)=\left|x^{(2)}\oplus x_1\right|=|000110\oplus 101101|=|101011|=4$$

$$\delta\left(x^{(3)},x_1\right)=\left|x^{(3)}\oplus x_1\right|=|010010\oplus 101101|=|111111|=6$$

$$\delta\left(x^{(4)},x_1\right)=\left|x^{(4)}\oplus x_1\right|=|010100\oplus 101101|=|111001|=4$$

$$\delta\left(x^{(5)},x_1\right)=\left|x^{(5)}\oplus x_1\right|=|100101\oplus 101101|=|001000|=1$$

$$\delta\left(x^{(6)},x_1\right)=\left|x^{(6)}\oplus x_1\right|=|100011\oplus 101101|=|001110|=3$$

$$\delta\left(x^{(7)},x_1\right)=\left|x^{(7)}\oplus x_1\right|=|110111\oplus 101101|=|011010|=3$$

$\delta\left(x^{(8)}, x_1\right) = \left|x^{(8)} \oplus x_1\right| = |110001 \oplus 101101| = |0111000| = 3$

$\therefore Min\, \delta\left(x^{(i)}, x_1\right) = \left(x^{(5)} \oplus x_1\right)$. Thus $x^{(5)}$ is the code word that is closest to x_1 and $e(011) = x^{(5)}$.

$\therefore$ We define maximum likelihood function d associated with e by $d(x_1) = 100$.

(2) Let $x_1 = 011011$

$\delta\left(x^{(1)}, x_1\right) = \left|x^{(1)} \oplus x_1\right| = |000000 \oplus 011011| = |011011| = 4$

$\delta\left(x^{(2)}, x_1\right) = \left|x^{(2)} \oplus x_1\right| = |000110 \oplus 011011| = |011101| = 4$

$\delta\left(x^{(3)}, x_1\right) = \left|x^{(3)} \oplus x_1\right| = |010010 \oplus 011011| = |001001| = 2$

$\delta\left(x^{(4)}, x_1\right) = \left|x^{(4)} \oplus x_1\right| = |010100 \oplus 011011| = |001111| = 4$

$\delta\left(x^{(5)}, x_1\right) = \left|x^{(5)} \oplus x_1\right| = |100101 \oplus 011011| = |111110| = 5$

$\delta\left(x^{(6)}, x_1\right) = \left|x^{(6)} \oplus x_1\right| = |100011 \oplus 011011| = |111000| = 3$

$\delta\left(x^{(7)}, x_1\right) = \left|x^{(7)} \oplus x_1\right| = |110111 \oplus 011011| = |101100| = 3$

$\delta\left(x^{(8)}, x_1\right) = \left|x^{(8)} \oplus x_1\right| = |110001 \oplus 011011| = |101010| = 3$

$\therefore Min\, \delta\left(x^{(i)}, x_1\right) = \left(x^{(3)} \oplus x_1\right)$. Thus $x^{(3)}$ is the code word that is closest to x_1 and $e(011) = x^{(3)}$.

$\therefore$ We define maximum likelihood function d associated with e by $d(x_1) = 010$.

Example 7.24 : Let $\mathrm{H} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ be a parity check matrix.

Decode the following words relative to a maximum likelihood decoding function associated with $\mathrm{e_H}$: (i) 011001, (ii) 101001, (iii) 111010.

Example 7.25 : Let $\mathrm{H} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ be a parity check matrix. [Nov-06]

Determine the $(3, 6)$ encoding function $\mathrm{e_H : B^3 \to B^6}$. Decode the words 011001 relative to a maximum likelihood decoding function associated with $\mathrm{e_H}$.

Solution : Let $\mathrm{e_H : B^3 \to B^6}$

$\mathrm{B^3} = \{000, 001, 010, 011, 100, 101, 110, 111\}$

$e_H(000) = 000000 = x^{(1)}, \quad e_H(001) = 001011 = x^{(2)},$

$e_H = (010) = 010110 = x^{(3)}, \; e_H = (011) = 011101 = x^{(4)},$

$e_H = (100) = 100100 = x^{(5)}, \; e_H = (101) = 101111 = x^{(6)},$

$e_H = (110) = 110010 = x^{(7)}, \; e_H = (111) = 111001 = x^{(8)}$

(i) Let $x_1 = 011001$

$$\delta\left(x^{(1)}, x_1\right) = \left|x^{(1)} \oplus x_1\right| = |0110001| = 3$$

$$\delta\left(x^{(2)}, x_1\right) = \left|x^{(2)} \oplus x_1\right| = |010010| = 2$$

$$\delta\left(x^{(3)}, x_1\right) = \left|x^{(3)} \oplus x_1\right| = |001111| = 4$$

$$\delta\left(x^{(4)}, x_1\right) = \left|x^{(4)} \oplus x_1\right| = |00100| = 1$$

$$\delta\left(x^{(5)},x_1\right)=\left|x^{(5)}\oplus x_1\right|=|111101|=5$$

$$\delta\left(x^{(6)},x_1\right)=\left|x^{(6)}\oplus x_1\right|=|110110|=4$$

$$\delta\left(x^{(7)},x_1\right)=\left|x^{(7)}\oplus x_1\right|=|101011|=4$$

$$\delta\left(x^{(8)},x_1\right)=\left|x^{(8)}\oplus x_1\right|=|1000000|=1$$

$\therefore Min\,\delta\left(x^{(i)},x_1\right)=\left(x^{(4)}\oplus x_1\right)$. Thus $x^{(4)}$ is the code word that is closest to x_1 and $e(011)=x^{(4)}$.

$\therefore$ We define maximum likelihood function d associated with e by $d\left(x_1\right)=011$.

(ii) Let $x_1=101001$

$$\delta\left(x^{(1)},x_1\right)=\left|x^{(1)}\oplus x_1\right|=|101001|=3$$

$$\delta\left(x^{(2)},x_1\right)=\left|x^{(2)}\oplus x_1\right|=|100010|=2$$

$$\delta\left(x^{(3)},x_1\right)=\left|x^{(3)}\oplus x_1\right|=|111111|=6$$

$$\delta\left(x^{(4)},x_1\right)=\left|x^{(4)}\oplus x_1\right|=|110100|=3$$

$$\delta\left(x^{(5)},x_1\right)=\left|x^{(5)}\oplus x_1\right|=|001101|=3$$

$$\delta\left(x^{(6)},x_1\right)=\left|x^{(6)}\oplus x_1\right|=|000110|=2$$

$$\delta\left(x^{(7)},x_1\right)=\left|x^{(7)}\oplus x_1\right|=|011011|=4$$

$$\delta\left(x^{(8)},x_1\right)=\left|x^{(8)}\oplus x_1\right|=|010000|=1$$

$\therefore Min\,\delta\left(x^{(i)},x_1\right)=\left(x^{(8)}\oplus x_1\right)$. Thus $x^{(8)}$ is the code word that is closest to x_1 and $e(111)=x^{(8)}$.

$\therefore$ We define maximum likelihood function d associated with e by $d\left(x_1\right)=111$.

Example 7.26 : Let $H = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ be a parity check matrix.

Decode the following words relative to a maximum likelihood decoding function associated with e_H : (i) 10100, (ii) 01101, (iii) 11011.

Solution : Let $e_H : B^2 \to B^5$ where $B^2 = \{00, 01, 10, 11\}$

$e_H(00) = 00000, \quad e_H(01) = 01101, \quad e_H = (10) = 10011, e_H(11) = 11110$

Use the above decoding procedure.

Example 7.27 : Consider the $(2, 9)$ encoding function e defined by

$e(00) = 000\,000\,000, \quad e(01) = 011\ 101\ 100$

$e(10) = 101\ 110\ 001, \quad e(11) = 110\ 001\ 111$

Let d be an associated maximum likelihood function. How many errors will (e, d) correct.

Solution :

Let $x^{(1)} = 000\ 000\ 000$, $x^{(2)} = 011\ 101\ 100$, $x^{(3)} = 101\ 110\ 001$, $x^{(4)} = 110\ 001\ 111$.

$\oplus$	000 000 000	011 101 100	101 110 001	110 001 111
000 000 000	-	011 101 100	101 110 001	110 001 111
011 101 100		-	110 011 101	101 100 011
101 110 001			-	011 111 110
110001111				

$\therefore$ Minimum distance = 5 $\qquad \therefore 2k + 1 = 5 \qquad \therefore k = 2$

$\therefore (e, d)$ can correct $k = 2$ or fewer errors.

7.6 UNIT END EXERCISE

(1) Define the following. [Dec-02]

(i) Hamming Distance,

(ii) Minimum distance of an encoding function

(iii) Group Code

(iv) Decoding function

(2) Consider the $(2,6)$ encoding function e :

$e(00)=000000$ $e(01)=011110$ $e(10)=101010$ $e(11)=111000$

Find the minimum distance (ii) How many errors will e detect.

[May-03]

3) Let $H=\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ be a panty check matrix. decode the following words related to maximum likelihood technique associated with e_H @ a) 10100 b) 01101 c) 11011

4) Consider the (2, 4) group encoding function $e: B^2 \rightarrow B^4$ defined by
e(00) = 0000 e(10) = 1001
e(01) = 0111 e(11) = 1111

Decode the following words relative to a maximum likelihood decoding function a) 0011 b) 1011 c) 1111.

5) Consider the (3, 5) group encoding function $e: B^3 \rightarrow B^5$ defined by
e(000) = 0000 e(100) = 10011
e(001) = 00110 e(101) = 10101
e(010) = 01001 e(110) = 11010
e(011) = 01111 e(111) = 11100

Decode the following words relative to a maximum likelihood decoding function. a) 11001 b) 01010 c) 00111.

❖❖❖❖

8

CLASSIFICATION OF LANGUAGE

[Syllabus : Classification of Languages : Overview of Languages, Representation of regular languages and grammars, Finite state machines.]

Unit Structure

8.0 OBJECTIVES:

To study there types of structure used in models of computation, namely, grammar, finite-state machines & moore machine. Which will help us to understand computer science and data networking.

8.1 INTRODUCTION:

Computer can perform many tasks. Given a task, two questions arise. The first is : can it be carried out using a computer? Once we know that this first question has an affirmative answer, we an ask the second question : How can the task be carried out? Models of computation are used to help answer these questions.

Example 8.1 : Let A = {Sachin, Saurav, Virat, well, runs, fields, quickly, slowly}.

Then A* contains real sentences such as “Sachin runs quickly” and “Virat fields well” as well as nonsense sentences such as “Quickly Sachin well Virat Slowly”.

Here we separate the elements in each sequence with spaces. It is often done when the elements of A are words.

Regular Expression : The idea of a recursive formula for a sequence is useful in more general strings. In the formal languages and the finite state machines the concept of regular expression plays on important role.

A regular expression over A is a string constructed from the elements of A and the symbols $(,), \vee, *, \wedge$ according to the following definition.

1. The symbol $\wedge$ is a regular expression.
2. If $p \in$A, the symbol p is a regular expression.
3) If x and y are regular expressions, then the expression xy is regular.
4) If x and y regular expressions, then the expression $x \vee y$ is regular.
5) If x is a regular expression, then x* is regular.

Example 8.2 : Let A = {0,1}. Show that following expressions are all regular expressions over A. a) $0^*(0 \vee 1)^*$ b) $00^*(0 \vee 1)*1$ c) $(01)*(01 \vee 1)*$

8.2 STRINGS AND REGULAR EXPRESSION:

Given a set A, we can construct the set A* consisting of all finite sequences of elements of A. Often the set A is not a set of numbers, but some set of symbols. In this case, A is called an alphabet and the finite sequences in A* are called words from A, or sometimes strings from A.

For this case in particular, the sequences in A* are not written with commas. We assume that A* contains empty sequence or empty string, containing no symbols, and we denote this string by $\wedge$.

Catenation :

If $w_1 = s_1,s_2....s_n$ and $w_2 = t_1,t_2.....t_k$ are elements of A* for some set A, we define the catenation of w_1, and w_2 as the sequence $s_1,s_2.....s_n$ $t_1,t_2.....t_k$. The catenation of w_1, with w_2 is written as $w_1,.w_2$ or w_1w_2 and is another element of A*. Note that if $w \in A^*$ then $w^*\wedge = w$ and $\wedge^*w = w$. This property is convenient and is one of the main reasons for defining the empty string $\wedge$.

Solution : We know from definition of regular expression that -

a) From (2) $0,1 \in A \Rightarrow 0 \,\&\, 1$ are regular expression.
From (4) 0, 1 are regular expression $\Rightarrow 0 \vee 1$ is regular.
From (5) $0^{*} \,\&\, 0 \vee 1$ are regular $\Rightarrow 0^{*} \,\&\, (0 \vee 1)^{*}$ are regular.
From (3) $0^{*} \,\&\, (0 \vee 1)^{*}$ are regular $\Rightarrow 0^{*} \,\&\, (0 \vee 1)^{*}$ is regular.

b) We know that 0, 1 & $0^{*}\,(0 \vee 1)^{*}$ are regular.
From (3) using twice $\Rightarrow 00*(0 \vee 1)^{*}1$ must be regular.

c) From (3) 01 is regular expression.
From (4) 1^{*} & $\left(01 \vee 1^{*}\right)$ are regular.
From (3) $(01)^{*}\left(01 \vee 1^{*}\right)$ is regular.

8.3 REGULAR SETS :

Associated with each regular expression over A, there is a corresponding subset of A*. Such sets are called regular subsets of A* or just regular sets if no reference to A is needed.

To compute the regular set corresponding to a regular expression, we use the following rules.

1. The expression $\wedge$ corresponds to the set $\{\wedge\}$, where $\wedge$ is the empty string in A*.
2. If $x \in A$, then the regular expression x corresponds to the set {x).
3. If α and β are regular expressions corresponding to the subsets M and N of A*, then the expression $\alpha\beta$ corresponds to $M.N = \{s.t/s \in M \text{ and } t \in N\}$. Thus MN is a set of all catenations of strings in M with strings in N.
4. If the regular expressions α and β correspond to the subset M and N of A, then $\{\alpha \vee \beta\}$ corresponds to $M \cup N$.
5. If the regular expression α corresponds to the subset M of A* then $\{\alpha\}^*$ corresponds to the set M*. Note that M is a set of strings from A~ Elements from M* are finite sequences of such strings, and thus may themselves be interpreted as strings from A. Note also that we always have $\wedge \in M^*$.

Example 8.3: Let A={0, 1}. Find regular sets corresponding to the three regular expressions (a) $0^*(0 \vee 1)^*$ (b) $00^*\,(0 \vee 1)^*l$ (c) $(01)^*(01 \vee 1^*)$

Solution:

(a) The set corresponding to 0*(0∨1)* consists of al1 sequences of 0's and l's. Thus the set is A*.

(b) The expression 00*(0∨1)*1 corresponds to the set of all sequences of 0's and 1's that begin with at least one 0 and end with at least one 1.

(c) The expression (01)*(Olvl*) corresponds to the set of all sequences of 0's and 1's that either repeat O1 a total n ≥ 1 times, or begins with a total of n ≥ 0 repetitions of 01 and end with some number k ≥ 0 of 1's. This set includes, for example the strings 1111, 01, 0l0l01, 010101011111 and 011.

8.4 LANGUAGES :

Words in the English language can be combined in various ways. The grammar of English tells us whether a cornbination of words in a valid sentence.

For Example the peacock writes neatly is a valid sentence because it is formed from a noun phrase the peacock, followed by a erb phrase writes neatly. We do not care that it is meaningless. Since we are concerned only with the syntax and not with its semantics i.e. meaning.

Research in the automatic translation of one language to another has led to the concept of formal language. It is specified by a well-defined set of rules of syntax. Rules of syntax are important not only in linguistic but also in the study of programming languages.

Grammars - Languages can be specified in various ways. We describe important way to specify a language, namely, through the use of a grammar.

A grammar provides a set of symbols of various types and a set of rules for producing words. More precisely, a grammar has a vocabulary V, which is a set of symbols used to derive members of the language. Some of the elements of the vocabulary cannot be replaced by other symbols. These are called terminals, and other members of the vocabulary, which can be replaced by other symbols, are called non terminals. The set of terminals and non terminals are usually denoted by T and N respectively.

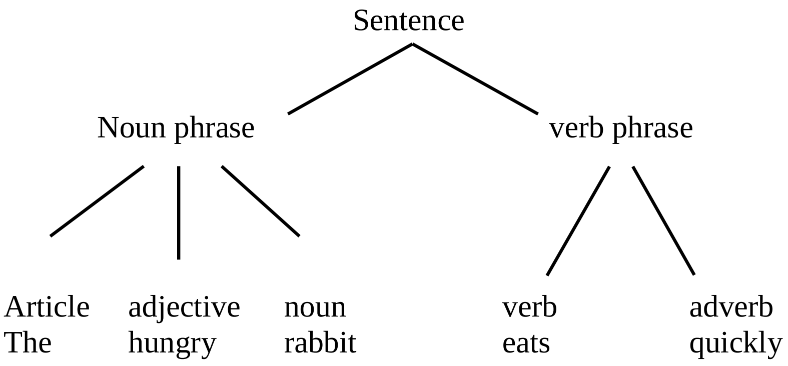
Definition : A phrase structure grammar G is defined to be a 4-type (V, S, $\vartheta_0, \mapsto$), where V is a finite set, S is a subset of $V, \vartheta_0 \in V - S$ and $\mapsto$ is a finite relation on V^*,. The element ϑ_0 is a starting point for the substitutions. The relation $\mapsto$ on V^* specifies allowable replacements. For example if $P \mapsto P'$, we may replace P by P'. Traditionally the statement

$P \mapsto P^{'}$ is called a Production of G. then P & P' are called the left and right sides of the production of G.

If $G=(V,S,\vartheta_0,\mapsto)$ is a phrase structure grammar, we call s the set of terminal symbols and N = V - S the set of non terminal symbols. Note that V = SUN.

Derivation Tree : a derivation in the language generated by a context - free grammar can be represented graphically using an ordered rooted tree, called a derivation tree. The root of this tree represents the starting symbol. The internal vertias of the tree represent the nonterminal symbols that arise in the derivation. The leaves of the tree represent the terminal symbols that arise.

For example - derivation tree for the derivation of "the hungry rabbit eats quickly" can be given as :



Language - The set of all properly constructed sentences that can be produced using a grammar G is called the language of G and is denoted by L (G)

Example 8.4: Let S = {Ramesh, Seema, drives, jogs; carelessly, rapidly, frequetitly}

N = {sentence, noun, verbphase, verb, adverb} and let $V = S \cup N$.

Let V_0 = sentence and suppose that the relation $\mapsto$ on V* is described by

sentence $\mapsto$ noun verbphrase
noun $\mapsto$ Ramesh
noun $\mapsto$ Seema
verbphrase $\mapsto$ verb adverb
verb $\mapsto$ drives
verb $\mapsto$ jogs
adverb $\mapsto$ carelessly
adverb $\mapsto$ rapidly
adverb $\mapsto$ frequently

The set S contains all the allowed words in the language; N consists of words that describe parts of sentences but that are not actually contained in the language. Write the derivation of the sentence “Seema drives rapidly.” Also draw the derivation tree.

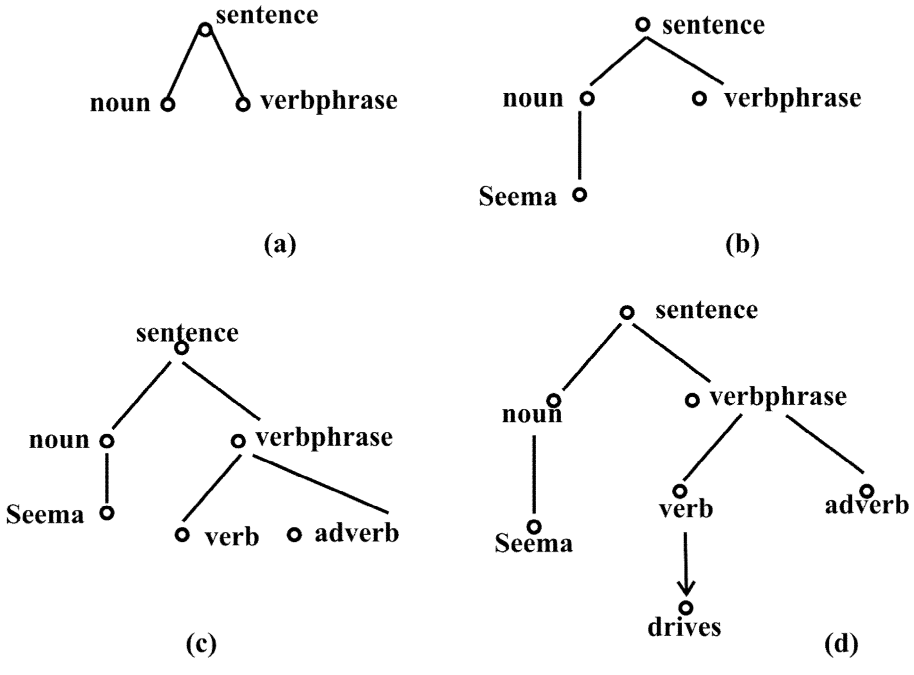
Solution : To prove this, we consider the following sequence of strings in V*

Sentence
noun verbphrase
Seema verbphrase
Seema verb adverb
Seema drives adverb
Seema drives rapidly.

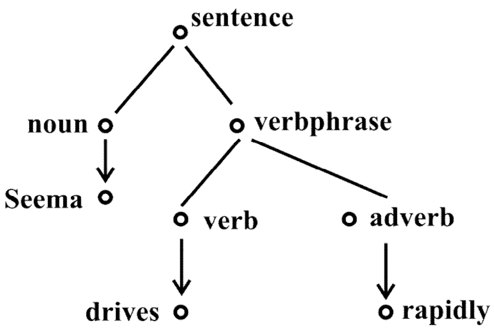
Note: Derivation of the sentence is not unique. Another derivation of the same sentence is given below.

sentence
noun verbphrase
noun verb adverb
noun verb rapidly
noun drives rapidly
Seema drives rapidly.

Derivation tree :



(a) (b) (c) (d)



Example 8.5 : Let V = {v_0, w, a, b, c} S = {a, b, c} and let $\mapsto$ be the relation on V* given by

1. $v_0 \mapsto$ aw 2. w $\mapsto$ bbw 3. w $\mapsto$ c

Consider the phrase structure grammar G = (V, s, v_0, $\mapsto$).

(i) Derive the sentence ab^6c. Also draw the derivation tree.

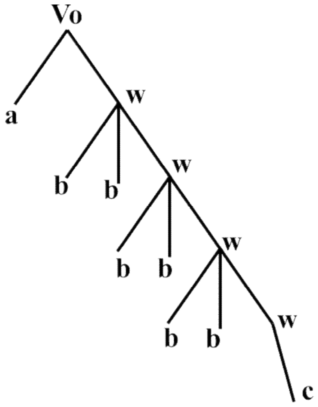
(ii) Derive the sentence ab^4c. Also draw the derivation tree. [Nov-06]

Solution :

(i) To drive sentences in L(G), it is necessary to perform successive substitutions, using (1), (2) and (3) until all symbols are eliminated other than the terminal symbols a, b and c.

v_0
aw
abbw
abbbbw
abbbbbbw
abbbbbbc
i.e. ab^6c

The derivation tree for ab^6c is shown below. It is not a binary tree :



Similarly we can draw derivation tree for ab^4c.

Example 8.6 : Let V = {v_0 w, a, b, c}, S = {a, b, c} and let $\mapsto$ be the relation on V* given by

1.vo $\mapsto$ av_0b 2. v_0b $\mapsto$ bw 3. abw $\mapsto$ c

Let G = (V, S, v_0, $\mapsto$) be the corresponding phase structure grammar. Determine the form of allowable sentences in L,(G). [May-06, May-07]

Solution: We may continue to use (1) any number of times, but we must eventually use production (2) to eliminate v_0.

Repeated use of (1) will result in a string of the form $a^n v_0 b^n$; i.e. there are equal number of a's and b's.

When (2) is repeatedly used, the result is a string of the form strings of the form a^m (abw) b^m with m ≥ 0

At this point the only production that can be used is (3).

$a^n cb^n \qquad n \geq 0.$

It cannot be expressed as trees.

Example 8.7

Determine whether the word cbab belongs to the language generated by the grammar $G = (V, S, v_0, \mapsto)$ where V = [a, b, c, A, B, C, S], T = [a, b, c], S is the starting symbol & the productions are

$S \mapsto AB$

$A \mapsto Ca$

$B \mapsto Ba$

$B \mapsto Cb$

$B \mapsto b$

$C \mapsto cb$

$G \mapsto b$

Solution : $S \mapsto AB$

$S \mapsto Cab$ by using $A \mapsto Ca$

$S \mapsto cbaB$ by using $C \mapsto ab$

$S \mapsto cbab$ by using $B \mapsto b$

∴ cbab belongs to the language generated by G. There are different approaches to get the result.

8.5 CLASSIFICATION OF PHRASE STRUCTURE GRAMMER

Let G = (V, S, v_0, $\mapsto$) be a phrase structure grammar. Then we say that G is

1. **Type 0:** if NO restrictions are placed on the productions of G.

2. **Type 1:** if for any production $w_1, \mapsto w_2$, the length of w, is less than or equal to the length of w_2. (where length of a string is the number of words in that string).

3. **Type 2:** if the left hand side of each production is a single, nonterminal symbol and the right hand side consists of one or more symbols.

4. **Type 3:** if the left hand side of each production is a single, nonterminal symbol and the right hand side consists of one or more symbols, including at most one nonterminal symbol, which must be at the extreme right of the string.

Note: In each of the preceeding types, we permit the inclusion of the trivial production $v_0 \mapsto \wedge$, where $\wedge$ represents the empty string.

It follows from the definition that each type of grammar is a special case of the type preceding it.

In the above illustrations example-4 is a type-2 grammar, example-5 is a type-3 grammar and example-4 is a type-0 grammar.

Grammar of type0 or 1 are quite difficult to study and little is known about them.

Context-free grammar: Type-2 grammers are sometimes called context-free grammar, since the symbols on the left of the productions are substituted for wherever they occur. A language generated by a type 2 grammar is called a context - Free language. When there is a production of the form $aw_1b \mapsto aw_2b$, the grammar is called type 1 or context - sensitive because w_1 can be replaced by w_2 only when it is surrounded by the strings a & b.

Regular Grammar : Type-3 grammers are also called regular grammar.

The process we have considered in this section mainly dividing a sentence within a grammar has a converse process. The converse process involves taking a sentence and verifying that it is syntactically correct in some grammar G by constructing a derivation tree that will produce it. This process is called parsing the sentences; and the resulting derivation tree is often called the parse tree of the sentence. Parsing is of fundamental importance for compilers and other forms of language translation. A sentence in one language is parsed to show its structure, and a tree is constructed. The tree is then searched and, at each step, corresponding sentences are generated in another language. In this way a C++ program, for example, is compiled into a machine language program.

8.6 REPRESENTATION OF SPECIAL GRAMMARS AND LANGUAGES :

There is another notation that is sometimes used to specify a type 2 grammar, called Backus - Naur Form (BNF), after John Backus, who invented it and Peter Naur who refined it for use in the specification of the programming language ALGOL. The Backus-Naur Form is used to specify the syntactic rules of many computer languages, including Java.

We know that the productions in type - 2 grammar have a single nonterminal symbol as their left-hand side. Instead of listing all the productions separately, we can combine all those with the same non terminal symbol on the left-hand side into one statement. Instead of using the symbol $\mapsto$ in a production we use the symbol = we enclose all nonterminal symbols in brackets, <>, and we list all the right-hand sides of productions in the same statement, separating them by bars. For example the production.

$A \mapsto Aa, A \mapsto a$ and $A \mapsto AB$ can be combined into A $\because = \langle A \rangle a \,|a|\, \langle A \rangle \langle B \rangle$.

Example 8.8 :

In BNF notation, the productions of example 4 appear as follows :

$\langle \text{sentence} \rangle \therefore \langle \text{noun} \rangle \langle \text{verb phrase} \rangle$

$\langle \text{noun} \rangle \therefore = \text{Ramesh / Seema}$

$\langle \text{verb phrase} \rangle \therefore = \langle \text{verb} \rangle \langle \text{adverb} \rangle$

$\langle \text{verb} \rangle \therefore = \text{drives / Jogs}$

$\langle \text{adverb} \rangle \therefore = \text{carelessly / rapidly / frequently}$

Note that the left-hand side of a production may also appear in one of the strings on the right-hand side.

Thus in the second line of Example 8 $\langle w \rangle$ appears on both sides. When this happens, we say that the corresponding production $w \mapsto bbw$ is recursive.

If a recursive production has w as left-hand side, we will say that the production is normal if w appears only one on the right-hand side and is the rightmost symbol. The recursive production $w \mapsto bbw$ is normal.

Example 8.9 :
Let V = {v_0, w, a, b, c} S = {a, b, c} and let $\mapsto$ be the relation on V* given by

1. $v_0 \mapsto aw$. 2. $w \mapsto bbw$ 3. $w \mapsto c$
Consider the phase structure grammar G = (V, S, v_0, $\mapsto$).
Write the production rules using BNF notations.

Solution: <v_0> :: = a < w >
<w> :: = bb < w > | c

Example 8.10 : BNF notation is often used to specify actual programming languages. PASCAL and many other languages had their grammars given in BNF initially. In this example we consider a small subset of PASCAL's grammar. This subset describes the syntax of decimal numbers and can be viewed as a mini-grammar whose corresponding language consists precisely of all properly formed decimal numbers.

Let S = {0, l, 2, 3, 4, 5, 6, 7, 8, 9, .}
Let V be the union of S with the set
N = {decimal-number, decimal-fraction, unsigned-integer, digit}

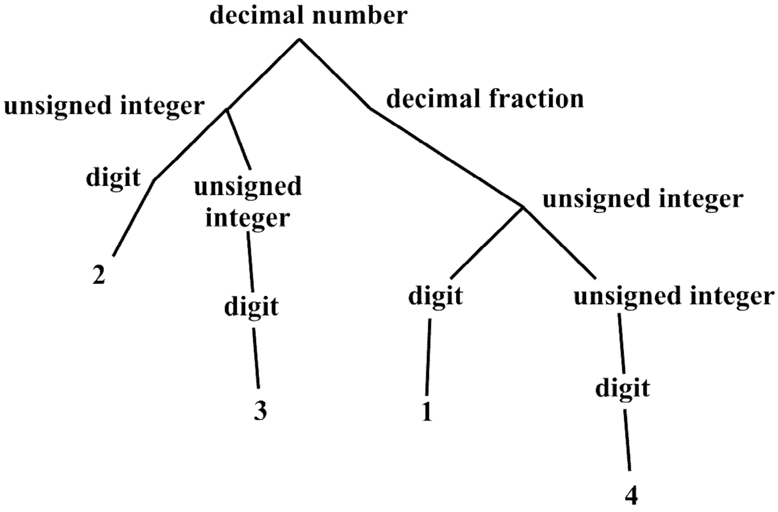
Let G be a grammar with symbol sets V and S, with starting symbol "decimal-number" and with productions given in BNF forms as follows.

1. < decimal-number > :: =
< unsigned-integer >|< decimal-fraction >|< unsigned-integer >
< decimal-fraction >
2. < decimal-fraction > :: - < unsigned-integer >
3. < unsigned-integer > :: = < digit > | < digit > < unsigned-integer >
4. < digit > :: = 0|1|2|3|4|5|6|7|8|9

Following derivation tree, in this grammar, shows the decimal number 23.14.

Note that the BNF statement numberd 3 is recursive, i.e. the production.

"unsigned integer $\mapsto$ digit unsigned integer" is recursive and also normal.



Example 8.11

As in example 9, we give a grammar that specifies a piece of several actual programming languages. In these languages, an identifier (a name for a variable, function, subroutine and so on) must be composed of letters and digits and must begin with a letter. The following grammar, with productions given in BNF, has precisely these identifiers as its language.

G = (V, S, identifier, ↦)
N = {identifier, remaining, digit, letter}
S = {a, b, …..,z, 0, l, 2, …..,9} .
V = N∪S

1. <identifier>:: = <letter> | <letter> <remaining>
2. <remaining>:: = <letter>|<digit>|<letter><remaining>|<digit> <remaining>
3. <letter> :: = a | b | c |…|z
4. <digit>:: = 0|1|2|3|4|S|6|7|8|9

Again we see that the production "remaining ↦ letter remaining" and "remaining ↦ digit remaining" in BNF statement 2 are recursive & normal.

8.6.2 Syntax Diagram: A second alternative method for displaying the production in some type-2 grammars is the syntax diagram.

Example 8.12 : Draw syntax diagrams representing following BNF statements.

(i) BNF statement that involves just a single production, such as <w>:: = < w_1> <w_2> <w_3> will result in the diagram shown in Figure (i)

(ii) If Terminal symbols circles or ellipse, syntax diagram is shown in Figure (ii)

<w>:: = <w$_1$> <w$_2$> | <w$_1$> a | bc<w$_2$>

(iii)Normal recursive production.

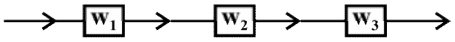
<w>:: = ab<w>

The syntax diagram for this production is shown in Figure (iii).

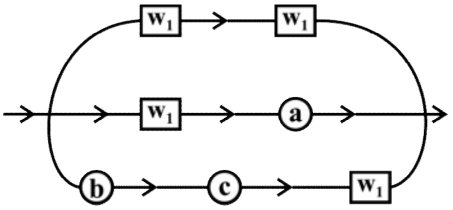
(iv) $\langle w \rangle \therefore = ab \vert ab \langle w \rangle$

The syntax diagram for this production is shown in figure (iv).

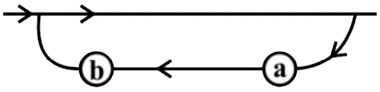
Solution :



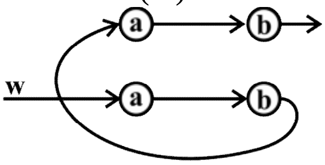
(i)



(ii)



(iii)



(iv)

Example 8.13 : For the grammar specified below describe precisely the language, L(G), produced. Also give the BNF and the corresponding syntax diagram for the productions of the grammar. [May-03, May-05]

G = (v,S,v$_0$, $\mapsto$)
V={v$_0$,a,b), S={a,b)
$\mapsto$: v$_0$ $\mapsto$ aav$_0$
v$_0$ $\mapsto$ a
v$_0$, $\mapsto$ b

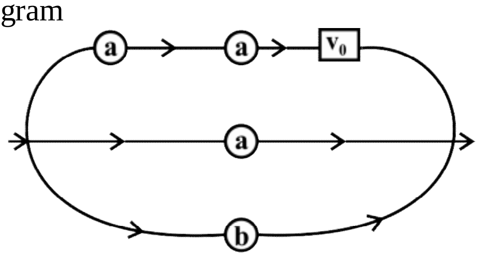
Solution: (i) L(G)

= {a2^{n+1}, n≥0} ∪ {a^{2n}b, n≥0}

(ii) BNF

<v$_0$>:: =. aa<v$_0$>~ | a | b

(iii) Syntax diagram



Example 8.14 : For the grammar specified below describe precisely the language, L(G), produced. Also give the BNF and corresponding syntax diagram for the productions of the grammar.

G = (v,S,v_0, $\mapsto$)
V = (v_0, v_1, x, y, z), S = {x, y, z}
: $v_0 \mapsto xv_0$
$v_0 \mapsto yv_1$,
$v_1 \mapsto yv_1$,
$v_1 \mapsto z$

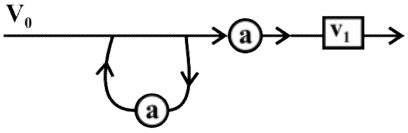
Solution: (i) L(G)
$x^n y^n z,\ \ m > 0, n \geq 1,$

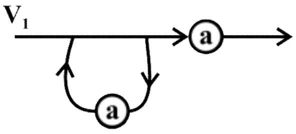
(ii) BNF

$< \mathrm{V}_0 > :: \ = x\langle \mathrm{V}_0\rangle | \mathrm{y}\langle \mathrm{V}_1\rangle$

$< \mathrm{V}_1 > :: \ = y\langle V_1\rangle |\ \mathrm{z}$

(iii) Syntax diagram





8.7 REGULAR GRAMMARS AND REGULAR EXPRESSION :

Theorem-1: Let S be a finite set, and L $\subseteq$ S*. Then L is a regular set if and only if L = L(G) for some regular grammar G = (V, S, v_0, $\mapsto$)

Theorem-1 tells us that the language L(G) of a regular grammar G must be the set corresponding to some regular expression over S, but it does not tell us how to find such a regular expression

8.8 FINITE STATE MACHINES :

Many kinds of machines, including components in computers, can be modeled using a structure called a finite-state machine. Several types of finite-state machines are commonly used in models. All these versions of finite-state machines include a finite set of states, with a designated starting state, an input alphabet and a transition function that assigns a next state to every state and input pair. Finite-state machines are used extensively in applications in computer science and data networking for example, finite - state machines are the basis for programs for spell checking, grammar checking, indexing or searching text, recognizing speech, transforming text using markup language such as HTML and network protocols that specify how computers communicate.

Suppose that we have a finite set S ={s_0,s_l, ... ,Sn] , a finite set I, and for each $x \in$I, a function f_x:S→S. Let F ={f_x,/x∈I}. The triplet (S,I,F) is called a finite state machine, S is called the state set of the machine and the elements of S are called states. The set I is called the input set of the machine. For any input $x \in$I, the function f_x describes the effect that this input has on the states of the machine and is called a state transition function. Thus, if the machine is in state si. and input x occurs, the next state of machine will be $f_x(s_i)$.

Since the next state fx(s_i) is uniquely determined by the pair (s_i, x) there is a function F:SXI→S given F(si, x) = fx(s_i)

The individual function, f_x can all be recovered from a knowledge of F.

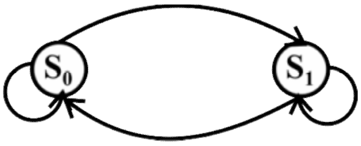
Example 8.15 : Let S={s_0, s_1} and I = {0, 1}. Define f_0 and f_1 as follows : $f_0(s_0) = s_0$, $f_0(s_1) = s_1$, $f_1(s_0) = s_1$ $f_1(s_1) = s_0$

This finite state machine has two states s_0s_1, and accept two possible inputs 0 and 1. Give transition table. Also draw diagraph of the finite state machine. [Nov-06]

Solution : We can summarize this machine as follows :

	0	1
s_0	s_0	s_1
s_1	s_1	s_0

(State Transition Table) This devise is often called as T flip-flop

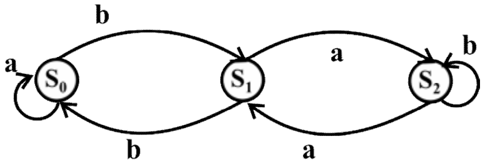


Example 8.16 : Consider the state transition table shown below

	a	b
s_0	s_0	s_1
s_1	s_2	s_0
s_2	s_1	s_2

Draw digraph of the machine

Solution :

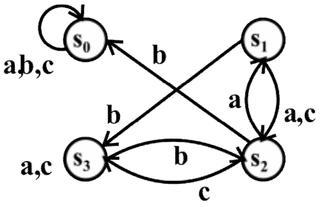


Example 8.17 : Consider the finite state machine M whose transition table is shown below

	a	b	c
s_0	s_0	s_0	s_0
s_1	s_2	s_3	s_2
s_2	s_1	s_0	s_3
s_3	s_3	s_2	s_3

Draw digraph of the machine

Solution :



8.9 MOORE MACHINE (RECOGNITION MACHINE) :

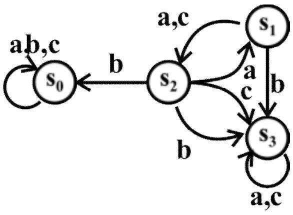
Many different kinds of finite-state machines have been developed to model computing machines. There is important type of finite-state machine with output, where the output is determined only by the state. This type of finite state machine is known as a Moor Machine, because E.F. moore introduced this type of machine in 1956.

It is defined as a sequence (S, I. F. s_0, T) where (S, I, F) constitute a finite state machine, $s_0 \in$ S and T $\subseteq$ S. The State s_0 is called the starting state of M, and it will be used to represent the condition of the machine before it receives any input. The set T is called the set of acceptance state of M. These states will be used in language recognition.

When the diagram of Moore machine is drawn, the acceptance states are indicated with two cocentric circles, instead of one. No special notation will be used on these diagraphs for the starting state, but unless otherwise specified, this state will be named so.

Example 8.18 : Let M be Moore machine (S, I. F, s_0, T) where (S, I, F) is the finite-state machine of figure in example 4 and T = {s_1, s_3}. Show the digraph of M.

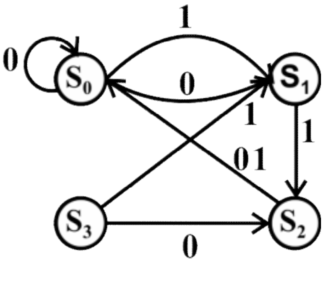
Solution : The diagraph of M is as follows :



Example 8.19 : Draw the diagraph of the machine whose state transition table is shown. Remember to label the edges with the appropriate inputs. M = (S, I, F, s_0, F) where S ={s_0, s_1, s_2, s_3}, I ={0, 1} and transition function is given in the table. [May-05]

State	Input	
	0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_0
s_3	s_2	s_1

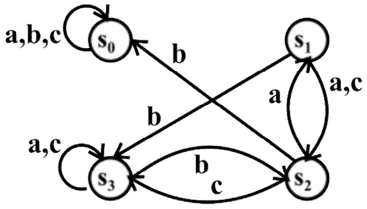
Solution : The state transition digraph is shown below.



Example 8.20 : Draw the state transition diagram for the following S = {s_0, s_1, s_2, s_3), I = {a, b; c}. [Dec-02, May-07]

	a	b	c
s_0	s_0	s_0	s_0
s_1	s_2	s_3	s_2
s_2	s_1	s_0	s_3
s_3	s_3	s_2	s_3

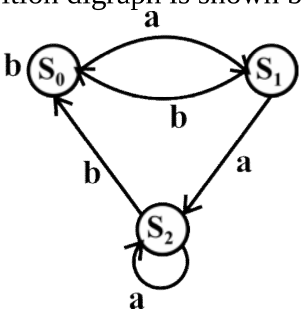
Solution: The state transition digraph is shown below.



Example 8.21 : Draw the diagraph of the machine whose state transition table is shown. Remember to label the edges with the appropriate inputs. [Dec-04]

	a	b
s_0	s_1	s_0
s_1	s_2	s_0
s_2	s_2	s_0

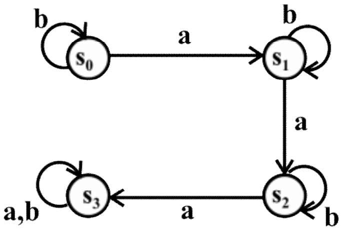
Solution : The state transition digraph is shown below.



Example 8.22 : Draw the diagraph of the machine whose state transition table is shown: [Oct-03]

	A	B
s_0	s_1	s_0
s_1	s_2	s_0
s_2	s_2	s_0

Solution: The state transition digraph is shown below.

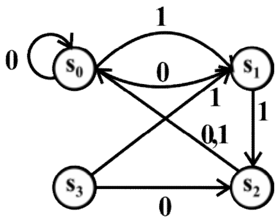


Example 8.23 : Draw the diagraph of the machine whose state transition table is shown. Remember to label the edges with the corresponding inputs.

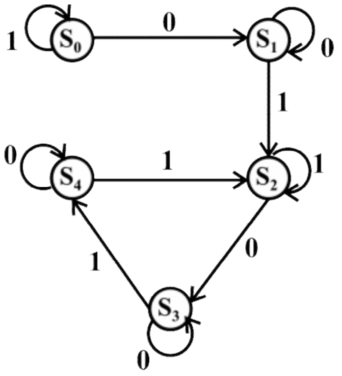
M = (S, I, F),where S = {s0, s1, s2, s3}, I ={0, 1} and the transition function is given below in table

State	Input	
	0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_0
s_3	s_2	s_1

Solution : The state transition digraph is shown below.



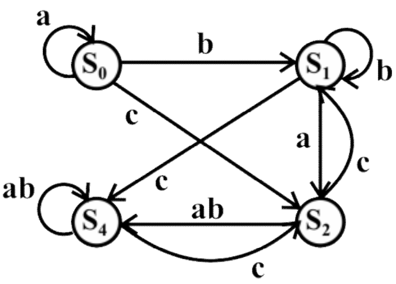
Example 8.24 : Construct the state transition table of the finite state machine whose diagraph is shown below.



Solution : The state transition digraph is shown below.

State	Input	
	0	1
S_0	S_1	S_0
S_1	S_1	S_2
S_2	S_3	S_2
S_3	S_3	S_4
S_4	S_4	S_2

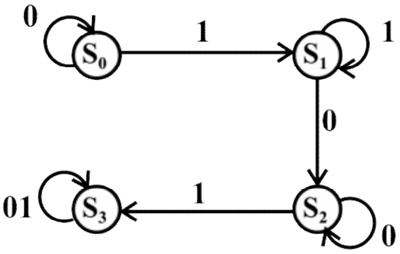
Example 8.25 : Construct the transition table of the finite state machine whose diagraph is. [Dec.-04, Nov-06]



Solution : State transition table of the given machine is shown below

State	**Input**		
	A	B	C
s_0	s_0	s_1	s_2
s_1	s_2	s_1	s_3
s_2	s_3	s_3	s_1
s_3	s_3	s_3	s_2

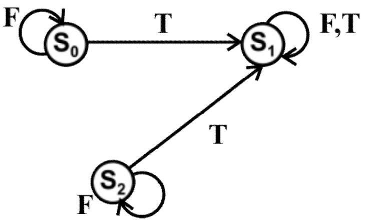
Example 8.26 : Construct the state transition table of the finite state machine whose diagraph is shown.. [Oct.-03, May-06]



Solution : State transition table of the given machine is shown below.

State	Input	
	0	1
S_0	S_0	S_1
S_1	S_2	S_1
S_2	S_2	S_3
S_3	S_3	S_3

Example 8.27: Construct the state transition table of the finite state machine whose diagraph is shown below. [Apr-04]



Solution: State transition table of the given machine is shown below.

State	Input	
	F	T
S_0	S_0	S_1
S_1	S_1	S_1
S_2	S_2	S_1

Example 8.28: Let the state transition table for a finite state machine be [Dec-02, Nov-05,May 06, May-07]

State	**Input**	
	0	1
S_0	S_0	S_1
S_1	S_1	S_2
S_2	S_2	S_3
S_3	S_3	S_0

List values of the transition function f_w for (i) w = 01001, (ii) w = 11100.

Solution: (i) w = 01001

$S_0 \xrightarrow{0} S_0 \xrightarrow{1} S_1 \xrightarrow{0} S_1 \xrightarrow{0} S_1 \xrightarrow{1} S_2 \qquad \therefore f_w(S_0) = S_2$

$S_1 \xrightarrow{0} S_1 \xrightarrow{1} S_2 \xrightarrow{0} S_2 \xrightarrow{0} S_2 \xrightarrow{1} S_3 \qquad \therefore f_w(S_1) = S_3$

$S_2 \xrightarrow{0} S_2 \xrightarrow{1} S_3 \xrightarrow{0} S_3 \xrightarrow{0} S_3 \xrightarrow{1} S_0 \qquad \therefore f_w(S_2) = S_0$

$S_3 \xrightarrow{0} S_3 \xrightarrow{1} S_0 \xrightarrow{0} S_0 \xrightarrow{0} S_0 \xrightarrow{1} S_1 \qquad \therefore f_w(S_3) = S_1$

(ii) w=1 1100

$$S_0 \xrightarrow{1} S_1 \xrightarrow{1} S_2 \xrightarrow{1} S_3 \xrightarrow{0} S_3 \xrightarrow{0} S_3 \qquad \therefore f_w(S_0) = S_3$$

$$S_1 \xrightarrow{1} S_2 \xrightarrow{1} S_3 \xrightarrow{1} S_0 \xrightarrow{0} S_0 \xrightarrow{0} S_0 \qquad \therefore f_w(S_1) = S_0$$

$$S_2 \xrightarrow{1} S_3 \xrightarrow{1} S_0 \xrightarrow{1} S_1 \xrightarrow{0} S_1 \xrightarrow{0} S_1 \qquad \therefore f_w(S_2) = S_1$$

$$S_3 \xrightarrow{1} S_0 \xrightarrow{1} S_1 \xrightarrow{1} S_2 \xrightarrow{0} S_2 \xrightarrow{0} S_2 \qquad \therefore f_w(S_3) = S_2$$

Example 8.29: S={0, 1, 2, 3, ..., 9}
N={<deo-num>,<dec-frac>,<unsigned int>,<digit>}
<deo-num>::=<unsigned int>/<dec-frac>/<unsigned int><dec-frac>
<dec-num>::=<unsigned int>
<unsigned int>::=<digit>/<digit><unsigned int>
<digit>::0/l/2/..../9.

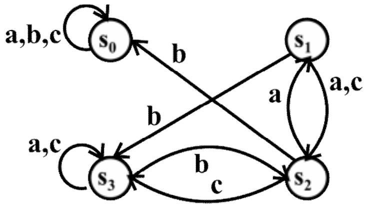
Give derivation tree and syntax diagram to represent decimal numbers using the above grammar. [Dec-02]

Example 8.30: Consider a Moore machine (S,I,F,s_0,T) where (S,I,F) is a finite state machine given by [Apr-04]

	a	b	c
S_0	S_0	S_0	S_0
S_1	S_2	S_3	S_2
S_2	S_1	S_0	S_3
S_3	S_3	S_2	S_3

And T={s_1,s_3}. Draw the digraph of the Moore machine.

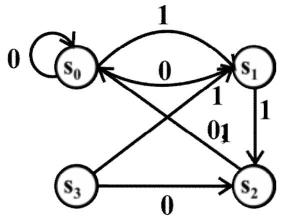
Solution: The digraph of the given Moore machine is shown below.



Example 8.31 : Define finite state automaton. Construct the state diagram for the finite-state automaton M = (S,I,s_0,F) where S = {s_0, s_1, s_2, s_3}; I = {0,1}, F = {s_0, s_3} and the transition function f is given in the table. [May-03]

State	**f**	
	Input	
	0	1
S_0	S_0	S_1
S_1	S_0	S_2
S_2	S_0	S_0
S_3	S_2	S_1

Solution: The state diagram for the finite state automaton is shown below.



8.10 UNIT END EXERCISE :

11) Construct a finite state machine that gives a 1 as its output bit if and only if the last three bits received are all 1's. [Oct-03]

12) Let M(S,I,F) be a finite state machine. Define a relation R on I as follows. [Oct-03]
x, Rx_2 if and only if f_{x1} (s)=f,$_{x2}$(s) for every s $\in$ S.
Show that R is an equivalence relation on I

8.11 REFERENCES :

1) Let A = [+, ×, a, b] show that the following expressions are regular over A.

i) $a+b(ab)^{*}(a\times b\vee a)$

ii) $a+b\times\left(a^{*}\vee b\right)$

iii) $\left(a^{*}b\vee+\right)^{*}\vee+b^{*}$

2) Let A = [a, b, c]. Give the regular set corresponding to the regular expression given i) $(a\vee b)cb^{*}$ ii) $a(bb)^{*}c$

3) Let S = [0,1]. Give the regular expression corresponding to the regular set given

i) [00, 010, 0110, 011110, ---]
ii) [0, 001, 000, 00001, 00000, 0000001, --]

4) Draw the diagraphs of the machines whose state transition table is shown below :

a)

State	Input	
	F	T
s_0	s_0	s_1
s_1	s_2	s_1
s_2	s_0	s_2

b)

	0	1	2
s_0	s_0	s_2	s_1
s_1	s_1	s_3	s_2
s_2	s_2	s_1	s_3
s_3	s_3	s_3	s_2

5) Let I = [0, 1] and S = [a, b]. construct all possible state transition tables of finite - state machines that have S as state set and I as input set.

6) Let a be the grammar with vocabulary V = [S, A, a, b], T = [a, b], starting symbol s, Productions $P=\left[S\mapsto aA, S\mapsto b, A\mapsto aa\right]$. What is L(a), the language of this grammar?

7) Give a phrase - structure grammar that generates the set $\left[0^n1^n \middle| n=0,1,2--\right]$

8) Give the BN form for the production of signed integers in decimal notation. (A signed integer is a nonnegative integer preceded by a plus sign or minus sign).

9) Let G be a grammar with V = [a, b, c, s], T = [a, b, c] starting symbol S and production $S\mapsto abs, S\mapsto bcs, S\mapsto bbs, s\mapsto a$ and $s\mapsto cb$. Construct derivation trees for i) bcbba ii) bbbcbba iii) bcabbbbbcb.

10) Find a phrase - structure grammar for each of these languages -

a) The set consisting of the bit strings 0, 1 & 11.
b) The set of bit strings containing only 1s.
c) The set of bit strings that start with 0 and end with 1.
d) The set of bit strings that consist of a 0 followed by an even number of 1s.

❖❖❖❖