SRM IST RAMAPURAM
DEPARTMENT OF MATHEMATICS

Sub. Code: 18MAB302T
Sub. Tittle: Discrete Mathematics for Engineers

S.No		Learning Unit/Module 4	
S-1	SLO-1	Binary operation on a set- Groups and axioms of groups.	1,2
	SLO-2	Properties of groups.	3
S-2	SLO-1	Permutation group, equivalence classes with addition modulo m and multiplication modulo m.	10
	SLO-2	Cyclic groups and properties.	16
S-3	SLO-1	Subgroups and necessary and sufficiency of a subset to be a subgroup.	18
	SLO-2	Group homomorphism and properties. Subgroups and necessary and sufficiency of a subset to be a subgroup.	19
S-4	SLO-1	Problem solving using tutorial sheet 10	20
	SLO-2	Problem solving using tutorial sheet 10	21
S-5	SLO-1	Rings- definition and examples..Zero devisors.	25
	SLO-2	Integral domain- definition , examples and properties.	25
S-6	SLO-1	Fields – definition, examples and properties.	26
	SLO-2	Coding Theory – Encoders and decoders- Hamming codes. Fields – definition, examples and properties.	27
S-7	SLO-1	Hamming distance. Error detected by an encoding function.	28
	SLO-2	examples	28
S-8	SLO-1	Problem solving using tutorial sheet 11	
	SLO-2	Problem solving using tutorial sheet 11	29
S-9	SLO-1	Error correction using matrices.	29
	SLO-2	Error correction using matrices.	
S-10	SLO-1	Group codes-error correction in group codes- parity check matrix. Problems on error correction in group codes.	30
	SLO-2	Group codes-error correction in group codes- parity check matrix. Problems on error correction in group codes.	30
S-11	SLO-1	Procedure for decoding group codes.	31
	SLO-2	Procedure for decoding group codes.	
S-12	SLO-1	Problem solving using tutorial sheet 12 Applications of sets, relations and functions in Engineering.	
	SLO-2		

Group Theory

Introduction:

The study of group theory is important in Computer science, not only because of the variety of applications of Computing techniques that involve massive algebra, relational databases, programming languages, requires an algebraic perspective. This chapter covers not only group concepts, group codes, parity and generator matrices and decoding and error Correction.

Binary operation on a set

Let A be a non-empty set. An operation on A is a function $f: A \times A \longrightarrow A$ is called a binary operation on A.

A function $g: A \times A \times A \longrightarrow A$ is called ternary operation on A.

More generally,

An N-ary operation is a function from $A \times A \times \cdots \times A$ (n factors) into A.

Group :

Let $(G, *)$ be an algebraic system with '$*$' as a binary operation. Then $(G, *)$ is called a group if the following axioms hold:

1. closure : If $a \in G$, $b \in G$, then $a * b \in G$ $\forall\ a, b \in G$

2. Associative: If $a, b, c \in G$, then $a * (b * c) = (a * b) * c$ $\forall\ a, b, c \in G$

3. Identity : There exist an element $e \in G$, such that $a * e = e * a = a$, $\forall\ a \in G$

4. Inverse : For each $a \in G$, there exists an element $a^{-1} \in G$ such that $a * a^{-1} = a^{-1} * a = e$.

Abelian group:

A Group G is said to be abelian, if the Commutative law holds

Commutative law: If $a, b \in G$ then $a * b = b * a$ $\forall\ a, b \in G$

Abelian group is called as Commutative group.

properties of group

Let $(G, *)$ be a group. Then

1. Identity element of G is unique.
2. Inverse element of G is unique.

Proof:

1. Suppose e_1 and e_2 be two identity elements in G.

since e_1 is an identity, we have

$$e_1 * e_2 = e_2 * e_1 = e_2 \quad \ldots\ldots (1)$$

since e_2 is an identity, we have

$$e_2 * e_1 = e_1 * e_2 = e_1 \quad \ldots\ldots (2)$$

from (1) & (2), $e_1 = e_2$

$\therefore$ The identity element is unique.

2. Let a_1 and a_2 be two inverses of a

suppose a_1 is an inverse, then

$$a * a_1 = a_1 * a = e \quad \ldots\ldots (3)$$

suppose a_2 is an inverse, then

$$a * a_2 = a_2 * a = e \quad \ldots\ldots (4)$$

from (3) and (4), we get

$$a_1 = a_2$$

$\therefore$ The inverse element is unique.

Theorem:

Let G be a group and let $a, b \in G$. Then

(i) $(a^{-1})^{-1} = a$

(ii) $(a * b)^{-1} = b^{-1} * a^{-1}$

proof:

(i) Let e be an identity of G. Then

$$a * a^{-1} = a^{-1} * a = e$$

$$a^{-1} * a = e$$

Taking $(a^{-1})^{-1}$ on both sides

$$(a^{-1})^{-1} * (a^{-1} * a) = (a^{-1})^{-1} * e$$

$$((a^{-1})^{-1} * a^{-1}) * a = (a^{-1})^{-1} \quad \text{(Associative law)}$$

$$e * a = (a^{-1})^{-1} \quad \text{(Inverse law)}$$

$$a = (a^{-1})^{-1}$$

(ii) Consider a^{-1} and b^{-1} be an inverse element of a and b, then we have

$$a * a^{-1} = a^{-1} * a = e$$

$$b * b^{-1} = b^{-1} * b = e$$

To prove $b^{-1}a^{-1}$ is an inverse of ab

i.e $(a*b) * (b^{-1} * a^{-1}) = (b^{-1} * a^{-1}) * (a * b) = e$

$$(a*b) * (b^{-1} * a^{-1}) = a * (b * b^{-1}) * a^{-1}$$

$$= a * e * a^{-1}$$

$$= a * a^{-1}$$

$$= e$$

$$(b^{-1} * a^{-1}) * (a * b) = b^{-1} * (a^{-1} * a) * b$$
$$= b^{-1} * e * b$$
$$= b^{-1} * b$$
$$= e$$
$$\therefore (a * b)^{-1} = b^{-1} * a^{-1}$$

Theorem (Cancellation law)

If G is a group and a, b, c are elements of G, then $a * b = a * c \Rightarrow b = c$

proof:

Assume $a * b = a * c$

Take a^{-1} on both sides

$$a^{-1} * (a * b) = a^{-1} * (a * c)$$
$$(a^{-1} * a) * b = (a^{-1} * a) * c$$
$$e * b = e * c$$
$$b = c$$

Theorem:

Let G be a group and a, b are any two elements of a group G, then $a * x = b$ and $y * a = b$ have unique solutions in G.

proof:

Consider $a, b \in G \Rightarrow a^{-1}, b \in G$

$$a^{-1} * b \in G$$
$$a * (a^{-1} * b) = (a * a^{-1}) * b = e * b = b$$

So $a^{-1} * b$ is a solution of $a * x = b$

To show it is unique

let us take x_1 and x_2 are solutions of

$a * x = b$

Then $a * x_1 = b$ and $a * x_2 = b$

$$a * x_1 = a * x_2$$

$$\Rightarrow x_1 = x_2 \quad \text{(Cancellation law)}$$

Similarly we can show $y = b a^{-1}$ is the unique solution of $y * a = b$.

Theorem

If every element of a group $(G, *)$ is its own inverse, then G is abelian.

proof: Given $a^2 = e$, $\forall\ a \in G$

$$a * a = e$$

$$a^{-1} * (a * a) = a^{-1} * e$$

$$(a^{-1} * a) * a = a^{-1} * e$$

$$e * a = a^{-1} * e$$

$$a = a^{-1}$$

i.e., every element is inverse of itself.

Now, for $a, b \in G$

$$a * b = a^{-1} * b^{-1}$$

$$= (b * a)^{-1}$$

$= b * a$

$\therefore \quad a * b = b * a$

Thus, G is abelian.

Problem

For any group G, prove that G is abelian if $(a * b)^n = a^n * b^n$, $\forall\, a, b \in G$, where n is a positive integer.

Solution: since G is abelian, we have $a * b = b * a$ $\forall\, a, b \in G$

Let $(a * b)^n = a^n * b^n$

for $n = 1$ $\quad (a * b)^1 = a * b$ is true

for $n = 2$ $\quad (a * b)^2 = (a * b) * (a * b)$

$= a * (b * a) * b$

$= a * (a * b) * b$

$= (a * a) * (b * b)$

$= a^2 * b^2$

Let for $n = k$ is true

$(a * b)^k = a^k * b^k$

for $n = k+1$ $\quad (a * b)^{k+1} = (a * b)^k * (a * b)$

$= (a^k * b^k) * (a * b)$

$= a^k * (b^k * a) * b$

$= a^k * (a * b^k) * b$

$$= (a^k * a) * (b^k * b)$$

$$= a^{k+1} * b^{k+1}$$

Hence it is true for $n = 1, 2, 3, \ldots n$.

problem prove that $G = \{0, 1, 2, 3, 4, 5\}$ is a finite abelian group of order 6 w.r.to addition modulo 6.

Solution: Cayley table

$+_6$	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

G is closed w.r.to addition modulo 6.

The Composition $+_6$ is associative

$$a +_6 (b +_6 c) = (a +_6 b) +_6 c$$

0 is the identity element.

$$0 +_6 a = a +_6 0 = a$$

The inverses of 0, 1, 2, 3, 4, 5 are 0, 5, 4, 3, 2, 1 respectively

$$a +_6 b = b +_6 a \quad \forall\ a, b \in G$$

$\therefore$ G is a finite abelian group.

Problem

Show that the identity element is the only idempotent element in the group.

Solution: Let a be an element of a group G such that $a^2 = a$. Now to prove that $a = e$

$$a^2 = a$$
$$\Rightarrow a * a = a$$
$$\Rightarrow (a * a) * a^{-1} = a * a^{-1}$$
$$\Rightarrow a * (a * a^{-1}) = e$$
$$\Rightarrow a * e = e$$
$$\Rightarrow a = e$$

Semi group: If a non-empty set G satisfies Closure and associative laws, then G is called semigroup.

Monoid : If a non-empty set G satisfies closure, associative and identity laws, then G is called monoid.

Examples:

(i) $(N, +)$ is a semigroup, since $0 \notin N$

(ii) $(Z, -)$ is not a semigroup. since it is not associative in Z.

(iii) $(Z^+, \cdot)$ is monoid. since $a^{-1} \notin Z^+, \forall\, a \in Z^+$.

(iv) $(Z, +)$ is a group.

(v) $(Z, \cdot)$ is a monoid.

(vi) $(Z, +)$ is an abelian group.

(vii) $(Q, +)$ is an abelian group.

Permutation group

A permutation is a 1-1 mapping of a non-empty set A onto itself.

A group $(G, *)$ is called a permutation group on a non-empty set P if the elements of G are permutations of P and the operation '$*$' is the composition of two functions.

If $S = \{1, 2, \ldots n\}$, the permutation group is also called the symmetric group of degree n and denoted by S_n. The number of elements of S_n is $n!$.

Example: Let $S = \{1, 2, 3\}$ and $\phi = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$

then $\phi = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 3 & 1 \\ 3 & 1 & 2 \end{pmatrix}, \begin{pmatrix} 3 & 1 & 2 \\ 1 & 2 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 3 & 2 \\ 2 & 1 & 3 \end{pmatrix}, \begin{pmatrix} 2 & 1 & 3 \\ 3 & 2 & 1 \end{pmatrix},$

$\begin{pmatrix} 3 & 2 & 1 \\ 1 & 3 & 2 \end{pmatrix}$.

Equality of permutations:

Let f and g be two permutations defined on a non-empty set P. Then, $f = g$ iff $f(x) = g(x)$, $\forall x \in P$.

Example: Let $S = \{1, 2, 3, 4\}$ and let

$f = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 2 & 4 \end{pmatrix}$ $g = \begin{pmatrix} 4 & 1 & 3 & 2 \\ 4 & 3 & 2 & 1 \end{pmatrix}$

we see that

$f(1) = 3 = g(1)$ $\quad$ $f(3) = 2 = g(3)$

$f(2) = 1 = g(2)$ $\quad$ $f(4) = 4 = g(4)$

Thus, $f(x) = g(x)$, $\forall x \in \{1, 2, 3, 4\}$, which implies that $f = g$.

Permutation identity:

An identity permutation on S, denoted by I, is defined as $I(a) = a$, $\forall a \in S$

Example: Let $S = \{1, 2, 3, 4\}$. Then

$f = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$ is the identity permutation on S.

Composition of permutations:

Let us Consider f and g be two arbitrary permutations of like degree; given by

$$f = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ b_1 & b_2 & \cdots & b_n \end{pmatrix} \qquad g = \begin{pmatrix} b_1 & b_2 & \cdots & b_n \\ c_1 & c_2 & \cdots & c_n \end{pmatrix}$$

on a non-empty set A. Then the Composition of f and g is defined as

$$fog = \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & a_n \\ c_1 & c_2 & c_3 & \cdots & c_n \end{pmatrix}$$

Example: Find the Composition of the following two Permutations

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix} \qquad g = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix}$$

Solution:

$$f \circ g = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$$

$$g \circ f = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix}$$

Thus $f \circ g \neq g \circ f$

i.e., It is not Commutative

Example: Let $S_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$ $S_2 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$ $S_3 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$

Then

$$S_1 \circ (S_2 \circ S_3) = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \circ \left[\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \right]$$

$$= \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

$$(S_1 \circ S_2) \circ S_3 = \left[\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \right] \circ \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

Hence, $S_1 \circ (S_2 \circ S_3) = (S_1 \circ S_2) \circ S_3$

It is associative.

Inverse permutation

Since a permutation is 1-1, onto and hence it is invertible.

If $f = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ b_1 & b_2 & \cdots & b_n \end{pmatrix}$ then $f^{-1} = \begin{pmatrix} b_1 & b_2 & \cdots & b_n \\ a_1 & a_2 & \cdots & a_n \end{pmatrix}$

Example: Let $f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 5 & 4 \end{pmatrix}$, then f^{-1}?

Solution:

Let $f^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ x & y & z & u & v \end{pmatrix}$

Then, $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 5 & 4 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ x & y & z & u & v \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$

$\Rightarrow \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ y & z & x & v & u \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$

$\Rightarrow x = 3,\ y = 1,\ z = 2,\ u = 5,\ v = 4$

Hence, $f^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 2 & 5 & 4 \end{pmatrix}$

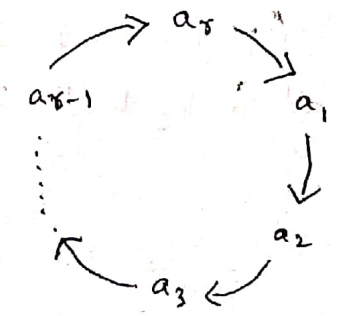
Cyclic permutations

Let $a_1, a_2, \ldots a_r$ be r distinct elements of the set $A = \{a_1, a_2, \ldots a_n\}$. Then the permutation $p : A \to A$ is defined by

$p(a_1) = a_2,\ p(a_2) = a_3, \ldots\ p(a_{r-1}) = a_r,\ p(a_r) = a_1.$

is called a cyclic permutation of length r.

It is denoted by $(a_1, a_2, \ldots a_r)$



Note: A permutation of a finite set, neither identity nor a cycle, can be expressed as a product of disjoint cycles of length ≥ 2.

Transposition: A cyclic permutation (a, b) which interchanges the symbols keeping all other intact is called a transposition.

Example:

$$(1\ 2\ 3\ 4\ 5) = (1\ 5) \circ (1\ 4) \circ (1\ 3) \circ (1\ 2)$$

note: Every permutation of 'n' symbols can be expressed as a product of disjoint cycles.

Even and odd permutations

A permutation is called an even permutation if it can be expressed as a product of an even number of transpositions, otherwise it is called as an odd permutation.

Example:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 6 & 2 & 4 & 1 & 3 \end{pmatrix} = (1\ 5)(2\ 6\ 3) = (1\ 5)(2\ 6)(2\ 3)$$

odd

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 3 & 4 & 5 & 2 & 1 \end{pmatrix} = (1\ 6)(2\ 3\ 4\ 5)$$

$$= (1\ 6)(2\ 3)(2\ 4)(2\ 5)$$

even

Note:

The set of all even permutations of degree n forms a finite group of order $\frac{n!}{2}$ w.r.to the Composition of permutation and is called alternating group. It is denoted by A_n.

problem: If z_6 is the set of equivalence classes generated by the equivalence relation "Congruence modulo 6", prove that $\{z_6, \times_6\}$ is a monoid.

Solution:

$\times_6$	[0]	[1]	[2]	[3]	[4]	[5]
[0]	0	0	0	0	0	0
[1]	0	1	2	3	4	5
[2]	0	2	4	0	2	4
[3]	0	3	0	3	0	3
[4]	0	4	2	0	4	2
[5]	0	5	4	3	2	1

The operation $\times_6$ is associative.

[1] is the identity element of $\{z_6, \times_6\}$.

Hence $\{z_6, \times_6\}$ is a monoid.

The elements [1] and [5] alone are invertible.

problem: Show that the set $\{z_m\}$ of equivalence classes modulo m is an abelian group under the operation $+_m$ of addition modulo m.

Solution: $z_m = \{[0], [1], [2], \ldots [m-1]\}$

Closure: Let $a, b \in z_m$ such that $a + b = q_1 m + r_1$ $0 \le r_1 < m$, then

$$[a] +_m [b] = [r_1] \in z_m$$

$\therefore z_m$ is Closed.

Clearly It is associative.

for every $[a] \in Z_m$

$$[a] +_m [0] = [0] +_m [a] = [a]$$

$[0]$ is an identity element

If $[a] \neq [0] \in Z_m$, then $[m-a] \in Z_m$ such that

$$[a] +_m [m-a] = [m] = [0]$$

Also $[m-a] +_m [a] = [0]$

$$[a]^{-1} = [m-a]$$

Inverse exists.

Now $[a] +_m [b] = [b] +_m [a] = [r_1]$

$\therefore$ Z_m is Commutative w.r.to $+_m$.

Thus $\{Z_m, +_m\}$ is an abelian group.

Cyclic groups

A group is called cyclic if for some $a \in G$, there exists element $x \in G$, of the form a^n, where n is some integer, The element 'a' is called a generator of G.

Example:

Let $G = \{1, -1, i, -i\}$ be a cyclic group.

properties

1. Every cyclic group is an abelian.

Let G be a cyclic group and let a be a generator of G so that

$$G = \langle a \rangle = \{ a^n : n \in \mathbb{Z} \}$$

Let a_1, a_2 be any two elements of G. Then, there exists some integers p and q such that $a_1 = a^p$ and $a_2 = a^q$

Now, $a_1 a_2 = a^p \cdot a^q = a^{p+q} = a^{q+p} = a_2 a_1$

$\Rightarrow$ G is an abelian group.

2. If a is a generator of a cyclic group G, then a^{-1} is also a generator of G.

Proof: Let $G = \langle a \rangle$ be a cyclic group generated by a.

Let a^p be any element of G, where p is some integer.

$a^p = (a^{-1})^{-p}$, here $-p$ is some integer

so, each element of G, is generated by a^{-1}.

Thus, a^{-1} is also a generator of G.

3. If a cyclic group G is generated by an element a of the order n, then a^m is a generator of G iff the g.c.d of m and n is 1.

Proof: Let a^m is a generator of $\{G, *\}$

Then, for some integer r,

$$a = (a^m)^r = a^{mr}$$

$$a = a^{mr} * e = a^{mr} * e^s \quad s \text{ some integer}$$

$$= a^{mr} * (e^n)^s$$

$$= a^{mr+ns}$$

$\therefore mr + ns = 1$

$GCD(m, n) = 1$

To prove the Converse, let $GCD(m,n) = 1$

$\therefore \exists$ two integers p and q such that

$$mp + nq = 1 \quad \ldots\ldots (1)$$

let H be the set generated by a^m.

$$H \subseteq G \quad \ldots\ldots (2)$$

Now $a^{mp+nq} = a$, by (1)

$$a^{mp} * a^{nq} = a$$

$$(a^m)^p * (a^n)^q = a$$

$$(a^m)^p * e = a$$

$$(a^m)^p = a$$

which means each integral power of a will also be an integral power of a^m.

$$G \subseteq H \quad \ldots\ldots (3)$$

from (2) and (3), we get $H = G$

a^m is a generator of G.

Subgroup:

Let $(G,*)$ be a group and H be a non-empty subset of G. If $(H,*)$ is itself a group, then $(H,*)$ is called a subgroup of $(G,*)$.

Example:

Let $G = \{1, -1, i, -i\}$ be a group.

$H = \{1, -1\}$ is a subgroup.

Note: $\{e\}$ and G are trivial subgroups.

All others subgroups are proper subgroups.

Theorem The necessary and sufficient condition for a non-empty subset H of a group G to be a subgroup is $a, b \in H \Rightarrow a * b^{-1} \in H$.

proof: **Necessary Condition**

Let H be a subgroup. Then if $a, b \in H$, $b^{-1} \in H$

$\therefore a * b^{-1} \in H$.

Sufficient Condition

Let $a * b^{-1} \in H$, where $a, b \in H$

If $a = b$, then $a * a^{-1} \in H \Rightarrow e \in H$

for $e, a \in H$, we have $e * a^{-1} \in H \Rightarrow a^{-1} \in H$

similarly, $b^{-1} \in H$

for a and $b^{-1} \in H$, we have

$a * (b^{-1})^{-1} \in H \Rightarrow a * b \in H$

$\therefore \{H, *\}$ is a group and hence a subgroup of G.

Group homomorphism:

If $\{G_1, *\}$ and $\{G_2, \Delta\}$ be two groups, then a mapping $f : G_1 \to G_2$ is called a group homomorphism, if for any $a, b \in G_1$, $f(a * b) = f(a) \Delta f(b)$.

A group homomorphism f is called group isomorphism, if f is 1-1 and onto.

Kernal of homomorphism:

If $f : G \to G'$ be a group homomorphism from G to G', then the set of elements of G, which are mapped into e', the identity element of G', is called the kernal of the homomorphism. It is denoted by $\ker(f)$.

Cosets: If $\{H,*\}$ is a subgroup of $\{G,*\}$, then the set aH, where $a \in G$, defined by

$$aH = \{a * h \mid h \in H\}$$

is called the left coset of H in G.

Similarly the set $Ha = \{h * a \mid h \in H\}$

is called the right coset of H in G.

Normal subgroup: A subgroup $\{H,*\}$ of the group $\{G,*\}$ is called a normal subgroup, if for any $a \in G$, $aH = Ha$.

Quotient (or) Factor group

If H is a normal subgroup of a group $(G,*)$ and G/H is the set of all cosets of H in G and if the binary operation $\otimes$ is defined by $aH \otimes bH = (a*b)H$, $\forall\ a,b \in G$, then $\{G/H, \otimes\}$ is a group called factor group.

Theorem: If H, K be two subgroups of a group G, then HK is a subgroup of G iff $KH = HK$.

proof: Let H, K be any two subgroups of G.

Let $HK = KH$

To prove HK is a subgroup of G

i.e to prove that $(HK)(HK)^{-1} = HK$

$$(HK)(HK)^{-1} = (HK)(K^{-1}H^{-1}) = H(KK^{-1})H^{-1}$$
$$= (HK)H^{-1}$$
$$= (KH)H^{-1}$$
$$= K(HH^{-1})$$
$$= KH = HK.$$

Conversely, HK is a subgroup.

Then $(HK)^{-1} = HK$

$\Rightarrow K^{-1}H^{-1} = HK$

$\Rightarrow KH = HK$ Hence the result.

Theorem: If H and K be two subgroups of G, then $H \cap K$ is also a subgroup of G.

proof: Let H and K be any two subgroups of G.

$H \cap K \neq \phi$ $\{e\}$ is common to both H and K.

To prove $H \cap K$ is a subgroup

$h \in H \cap K$, $k \in H \cap K \Rightarrow hk^{-1} \in H \cap K$

Now $h \in H \cap K \Rightarrow h \in H$ and $h \in K$

$k \in H \cap K \Rightarrow k \in H$ and $k \in K$

$h \in H, k \in H \Rightarrow hk^{-1} \in H$ $hk^{-1} \in H \cap K$

$h \in K, k \in K \Rightarrow hk^{-1} \in K$

Hence $H \cap K$ is a subgroup of G.

Theorem: The union of two subgroups of G is a subgroup iff one is contained in the other.

proof: Let H and K be subgroups of G.

Either $H \subseteq K$ or $K \subseteq H$

$\therefore H \cup K = H$ or $H \cup K = K$

Hence $H \cup K$ is a subgroup of G.

Conversely, suppose $H \cup K$ is a subgroup of G, then we claim $H \subseteq K$ or $K \subseteq H$.

Assume $H \not\subseteq K$ and $K \not\subseteq H$

then $\exists\, a, b$ such that $a \in H$ and $a \notin K$ and $b \in K$ and $b \notin H$

Clearly, $a, b \in H \cap K$

$ab \in H$ or $ab \in K$

$a \in H, a^{-1} \in H$

Hence $a^{-1}(ab) = b \in H$, which is a Contradiction.

Since $b \in K$, $b^{-1} \in K$

$(ab)b^{-1} = a \in K$ which is a Contradiction

Hence $H \subseteq K$ or $K \subseteq H$.

Theorem: (Fundamental theorem of homomorphism)

Let $f: G \to G'$ be a homomorphism. Let H be a kernel of f. Then $G/H \simeq G'$.

proof: Let $f: G \to G'$ and define $\phi: \frac{G}{H} \to G'$ as

$$\phi(Ha) = f(a)$$

i) **ϕ is well defined**

let $Hb = Ha$

$b = ha$ where $h \in H$

$$f(b) = f(ha) = f(h)f(a) = e'f(a) = f(a)$$

$$\therefore \phi(Hb) = f(b) = f(a) = \phi(Ha)$$

ii) **ϕ is 1-1**

$$\phi(Ha) = \phi(Hb)$$

$$\Rightarrow f(a) = f(b)$$

$$\Rightarrow f(a)(f(b))^{-1} = e'$$

$$\Rightarrow f(ab^{-1}) = e'$$

$$\Rightarrow ab^{-1} \in K$$

$$\Rightarrow a \in Hb$$

$$\Rightarrow Ha = Hb$$

iii) **ϕ is onto**

Let $a' \in G'$. since f is onto, $\exists\ a \in G$ such that $f(a) = a'$. Hence $\phi(Ha) = f(a) = a'$.

ϕ is a homomorphism

$$\phi(ha\,hb) = \phi(hab) = f(ab) = f(a)\,f(b) = \phi(ha)\,\phi(hb)$$

$\therefore G/H \simeq G'$.

Theorem: (Lagrange's)

The order of a subgroup of a finite group divides the order of the group.

proof: Let $G = \{a_1 = e, a_2, \ldots a_n\}$ be a group and H be a subgroup of G.

$$o(G) = n.$$

Consider
$$e * H = \{eh \mid h \in H\}$$
$$a_2 * H = \{a_2 h \mid h \in H\}$$
$$\vdots$$
$$a_n * H = \{a_n h \mid h \in H\}$$

we know any two left or right cosets are either identical or disjoint.

Also $o(e * H) = o(H)$

$\therefore o(a_i * H) = o(H) \quad \forall\, a_i \in G$

Let there be k-disjoint cosets of H in k and their union equals G.

$$G = (a_1 * H) \cup (a_2 * H) \cup \ldots \cup (a_k * H)$$

$$o(G) = o(a_1 * H) + \ldots + o(a_k * H)$$

$$= o(H) + \ldots + o(H) \quad (k \text{ times})$$

$$o(G) = k\, o(H)$$

$$k = \frac{o(G)}{o(H)}.$$

The order of a subgroup H divides the order of a group G.

Theorem: prove that intersection of two normal subgroups is a normal subgroup.

proof: Let H and K be any two normal subgroups of G.

since H and K are subgroups of G

$e \in H$ and $e \in K$ Hence $e \in H \cap K$.

Let $a, b \in H \cap K$

Claim: $ab^{-1} \in H \cap K$

since $a, b \in H \cap K$, both $a, b \in H$ and K.

since H and K are subgroups of G, $ab^{-1} \in H$ and K.

$ab^{-1} \in H \cap K$

Hence $H \cap K$ is a subgroup of G.

To prove: $H \cap K$ is normal

Let $x \in H \cap K$ and $g \in G$

since $x \in H \cap K \Rightarrow x \in H$ and $x \in K$

since $x \in H$, $g \in G \Rightarrow gxg^{-1} \in H$.

$x \in K$, $g \in G \Rightarrow gxg^{-1} \in K$

Hence $x \in H \cap K$ and $g \in G$

$\Rightarrow gxg^{-1} \in H \cap K$

Thus $H \cap K$ is a normal subgroup of G.

Ring:

$(R, +, \cdot)$ is called a ring if the binary operations $+$ and $\cdot$ on R satisfy the following conditions

i) $(R, +)$ is abelian

ii) $(R, \cdot)$ is a semigroup

iii) Distributive law: $a \cdot (b+c) = a \cdot b + a \cdot c \quad \forall a, b, c \in R$.

Example: $(=, +, \cdot)$, $(\nmid, +, \cdot)$, $(R, +, \cdot)$

Zero divisor:

R is Commutative ring, $a \neq 0 \in R$ is zero divisor if there exists $b \in R$, $b \neq 0$, $ab = 0$.

Integral domain

A Commutative ring is an Integral domain if it has no zero divisor.

Division ring (skew field)

A ring is called a division ring if its non-zero elements form a group under multiplication.

Field:

A Commutative division ring is called a field.

Example:

1. $(\mathbb{Z}_7, +_7, \cdot_7)$ is a field.
2. $(\mathbb{Z}_{10}, +_{10}, \cdot_{10})$ is not an Integral domain
3. $(\mathbb{Q}, +, \cdot)$, $(R, +, \cdot)$ are fields.
4. $(\mathbb{Z}, +, \cdot)$ is an integral domain but not a field.
5. M_n of all non-zero matrices is not Commutative. and has non-zero divisors.

Note:

1. A field is an integral domain.
2. A finite integral domain is a field.
3. Let R be a finite (non-zero) integral domain, then $o(R) = p^n$, where p is a prime.

Sub-ring

A non empty subset S of a ring R is said to be a subring of R if S forms a ring under the binary Compositions of R.

Example:

1. The ring $\langle z, +, \cdot \rangle$ of integers is a subring of the ring $\langle R, +, \cdot \rangle$ of real numbers.

2. The ring of integers $\langle z, +, \cdot \rangle$ is a subring of the ring of all rational numbers.

Ring homomorphism

A function $f : R \to R^*$ where R and R^* are rings is said to be a ring homomorphism if

(i) $f(a+b) = f(a) + f(b)$

(ii) $f(ab) = f(a) f(b) \quad \forall\ a, b \in R.$

Coding theory

Error detection and error correction techniques play an important role in the design of Computer systems. Structure in the design of error-correcting Codes is important. It makes easy in finding the properties of a Code and it makes to realize the hardware of such practical Codes.

Algebraic structures are the basis of the most important Codes which have been designed. A Communication process may take place in a variety of ways, by making a telephone Call, sending a message by a telephone or a letter, using a sign language, etc.

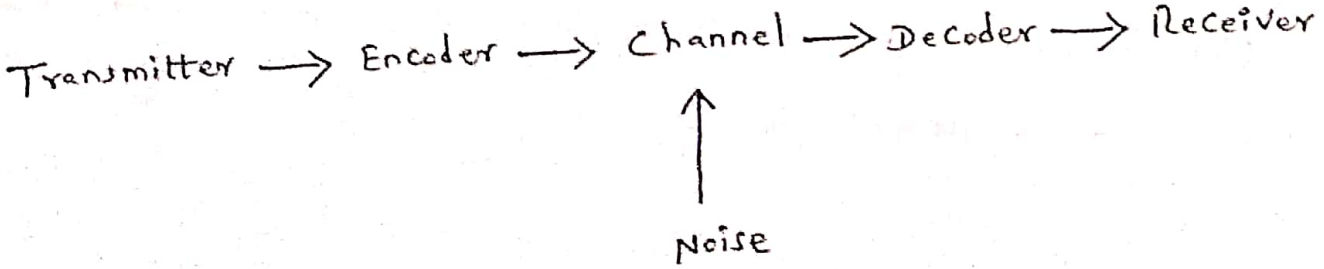
An ideal Communication system can be represented as follows

Transmitter ⟶ channel ⟶ Receiver

Encoders and Decoders

An encoder is a device which transforms the incoming messages in such a way that the presence of noise in the transformed messages is detectable.

A decoder is a device which transforms the encoded message into their original form that can be understood by the receiver. The model of a typical data Communication system with noise is given as



Group Code:

If $B = \{0, 1\}$, then $B^n = \{x_1, x_2, \cdots x_n \mid x_i \in B, i = 1, 2, 3, \cdots n\}$ is a group under the binary operation of addition modulo 2, This group $(B^n, \oplus)$ is called a group Code.

In general, any code which is a group under the operation $\oplus$ is called a group Code.

Hamming Codes: The codes obtained by introducing additional digits called parity digits to the digits in the original message are called Hamming Codes.

Hamming distance:

If x and y represent the binary strings $x_1, x_2, \cdots x_n$ and $y_1, y_2, \cdots y_n$, the number of positions in the strings for which $x_i \neq y_i$ is called the Hamming distance between x and y and denoted by $H(x, y)$.

Example: If $x = 11010$ and $y = 10101$, then

$$H(x, y) = |x \oplus y| = |01111| = 4.$$

Note: 1. A code can correct a set of at the most k errors iff the minimum distance between any two code words is at least $(2k+1)$.

2. A code can detect at the most, k errors iff the minimum distance between any two code words is at least $(k+1)$.

Error Correction using matrices

The encoding function $e: B^m \to B^n$, where $m, n \in z^+$ and $m<n$, where $B \equiv (0,1)$ is given by a $m \times n$ matrix G over B. This matrix G is called the generator matrix for the code and is of the form $[I_m / A]$, where I_m is the $m \times m$ unit matrix and A is an $m \times (n-m)$ matrix.

example If the message $w \in B^2$, we may assume G

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

The words that belong to B^2 are 00, 10, 01 and 11. Then the code words corresponding to the above message words are

$$e(00) = [0\ 0]\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [00\ 000]$$

$$e(10) = [1\ 0]\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [10\ 110]$$

$$e(01) = [0\ 1]\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [01011]$$

$$e(11) = [1\ 1]\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [11101]$$

parity and Generator matrices

The encoding function $e: B^m \to B^{m+1}$ is called the parity $(m, m+1)$ check code. If $b = b_1 b_2 \cdots b_m \in B^m$, define

$$e(b) = b_1 b_2 \cdots b_m b_{m+1}.$$

where

$$b_{m+1} = \begin{cases} 0, & \text{if } |b| \text{ is even} \\ 1, & \text{if } |b| \text{ is odd} \end{cases}$$

The weight of each of the following words in B^4

(i) $x = 0100 \Rightarrow |x| = 1$ (ii) $x = 1110 \Rightarrow |x| = 3$

(iii) $x = 0000 \Rightarrow |x| = 0$ (iv) $x = 1111 \Rightarrow |x| = 4$

(v) $x = 0110 \Rightarrow |x| = 2$

The number of 1's in x is called the weight of x and is denoted by $|x|$.

Decoding and error Correction

An onto function $D : B^n \to B^m$ is called an (n, m) decoding function associated with e, if $D(y) = x_t \in B^m$, and is such that when the transmission channel has no noise then $x_t = x$, i.e., $D \circ e = I$, where I is the identity function on B^m.

Example Let $e : B^3 \to B^4$ and $D : B^4 \to B^3$ is the decoding function

Code x	000	001	010	011	100	101	110	111
$y = e(x)$	0000	0011	0101	0110	1001	1010	1100	1110
$D(y) = (D \circ e)x$	000	001	010	011	100	101	110	111

problems: Find the code words generated by the encoding function $e : B^2 \to B^5$ w.r.to the parity check matrix

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

Solution:

Rewriting the given matrix as $H = [A^T / I_{n-m}]$

$$H = \left[\begin{array}{cc|ccc} 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Here $n = 5$ and $m = 2$

The generator matrix G is given by

$$G = [I_m / A] = \left[\begin{array}{cc|ccc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

Now $B^2 \equiv \{00, 01, 10, 11\}$ and $e(w) = wG$

$$e(00) = [0\ 0] \left[\begin{array}{ccccc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{array}\right] = [0\ 0\ 0\ 0\ 0]$$

$$e(01) = [0\ 1] \left[\begin{array}{ccccc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{array}\right] = [0\ 1\ 0\ 1\ 1]$$

$$e(10) = [1\ 0] \left[\begin{array}{ccccc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{array}\right] = [1\ 0\ 0\ 1\ 1]$$

$$e(11) = [1\ 1] \left[\begin{array}{ccccc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{array}\right] = [1\ 1\ 0\ 0\ 0]$$

Hence, the code words generated by H are

00000, 01011, 10011, and 11000.

problem Find the code words generated by the parity-check matrix

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

when the encoding function is $e : B^3 \longrightarrow B^6$

$$H = \left[\begin{array}{ccc|ccc} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right] = [A^T \mid I_{n-m}]$$

$n = 6,\ m = 3$

$$G = [I_m \mid A] = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

Now $B^3 = \{000, 001, 010, 100, 011, 101, 110, 111\}$

$e(000) = [000] \cdot G = [0\ 0\ 0\ 0\ 0\ 0]$

$e(001) = [001] \cdot G = [0\ 0\ 1\ 0\ 1\ 1]$

$e(010) = [010] \cdot G = [0\ 1\ 0\ 1\ 0\ 1]$

$e(100) = [100] \cdot G = [1\ 0\ 0\ 1\ 1\ 1]$

$e(011) = [011] \cdot G = [0\ 1\ 1\ 1\ 1\ 0]$

$e(101) = [101] \cdot G = [1\ 0\ 1\ 1\ 0\ 0]$

$$e(110) = [110] \cdot G = [110010]$$

$$e(111) = [111] \cdot G = [111001]$$

Thus, the code words are

000000, 001011, 010101, 100111, 011110, 101100, 110010 and 111001.

problem: Decode each of the following received words corresponding to the encoding function $e: B^3 \to B^6$ given by $e(000) = 000000$, $e(001) = 001001$, $e(010) = 010101$, $e(100) = 100111$, $e(011) = 011110$, $e(101) = 101100$, $e(110) = 110010$, and $e(111) = 111001$ assuming no error or signal error occured:

011110, 110111, 110000, 111000, 011111

Solution:

(i) The word 011110 is identical with $e(011)$. Hence no error has occured and original message is 011.

(ii) The word 110111 differs from $e(100) = 100111$ in the second position only. Correcting the single error, the transmitted word is 100111 and message is 100.

(iii) The word 110000 differs from $e(110) = 110010$ in the fifth position. Correcting the single error, the transmitted word is 110010 and original message is 110.

(iv) The word 111000 differs from $e(111) = 111001$ in the sixth position. Correcting this error, the transmitted word is 111001 and original 111.

(v) The word 011111 differs from $e(011) = 011110$ in the sixth position. Correcting the error, the transmitted word is 011110 and original is 011.

problem: Let $G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$ Corresponding to the encoding function $e: B^3 \to B^6$, find the parity Check matrix and use it to decode the following received words and find the original message. Are all the words decoded uniquely? $G = [I_3 | A]$

(i) 110101 (ii) 001111 (iii) 110001 (iv) 111111

Solution: $H = [A^T | I_3] = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$

Compute the syndrume of each of the received word by using $H \cdot [r]^T$.

(i) $H \cdot [r]^T = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

Since $H \cdot [e(w)]^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ The received word is the transmitted word itself. original message is 110.

(ii) $H \cdot [r]^T = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

since the syndrome $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ is the same as fifth Column of H, the element in the fifth position of r is Changed.

The decoded word is 0 0 1 1 0 1 and original is 0 0 1.

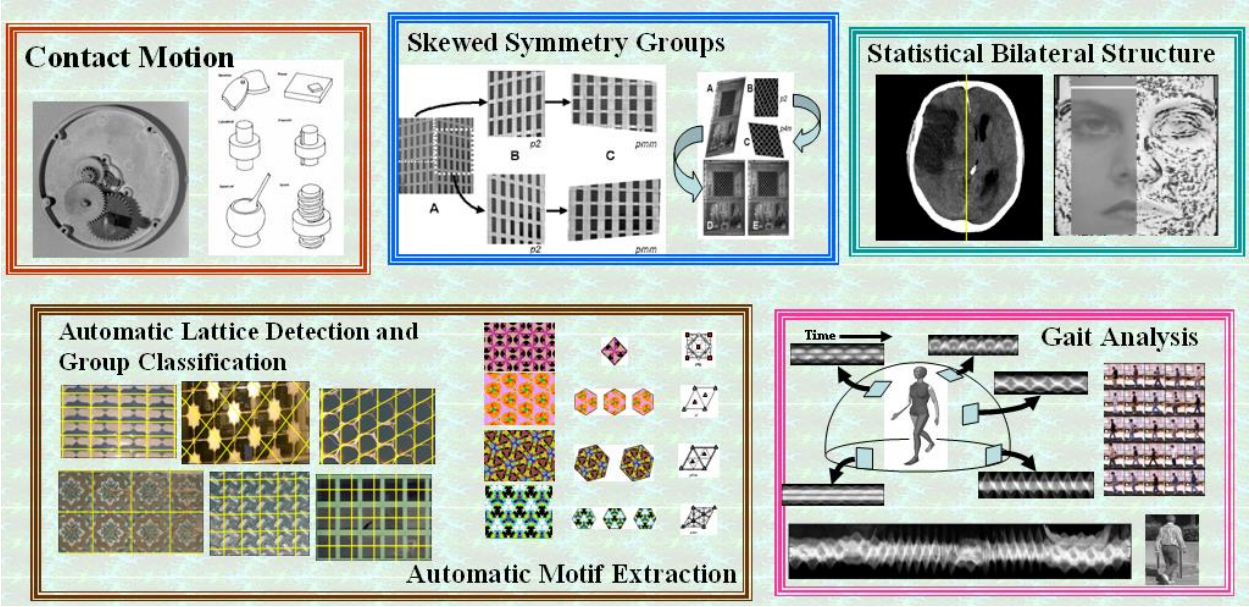
(iii) $H \cdot [r]^T = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

since the Syndrome $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ is the same as fourth Column of H the fourth position of r is Changed. The decoded word is 1 1 0 1 0 1 and original is 1 1 0.

(iv) $H \cdot [r]^T = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

Since, the syndrome is not identical with any Column of H, the received word Cannot be decoded uniquely.

Group Theory and Its Applications in Robotics, Computer Vision/Graphics and Medical Image Analysis



Description:

Group theory, the ultimate theory for symmetry, is a powerful tool that has a direct impact on research in robotics, computer vision, computer graphics and medical image analysis. This course starts by introducing the basics of group theory but abandons the classical definition-theorem-proof model. Instead, it relies heavily on intuitions in (1) 3D Euclidean space, images and patterns; (2) a geometric computational model; and (3) concrete, real world applications in robotics, computer vision, computer graphics and medical image analysis drawing from the instructor's many years of research experience and from an emerging, vibrant, interdisciplinary international research community. The material will be taught in a bottom-up (problems to theory) style based on the instructor's manuscript of "Group Theory Applications in Robotics, Computer Vision and Computer Graphics", state of art research papers and classical articles in prominent journals/books. The course emphasizes on motivations and justifications for the algorithmic usage of group theory in different domains, computational issues, and hands-on experimentation and illustration. The instructor encourages students to explore new applications while providing a handle on an elegant methodology and available computational tools. This course should be appropriate to any students who have an interest in real world problems that involve 3D Euclidean geometry, regularity, near-regular patterns and symmetry. It should be particularly attractive to students with computational inclinations of using algebraic theory in combination with other tools (e.g. graph theory, statistics). The goal is to provide the course material in a fairly high level of sophistication with intuition, formal justification and algorithmic ease.

JUSTIFICATION:

This course addresses both a real need in graduate education and in research communities of using formal methods in symmetry, asymmetry and near-regularity. Symmetry is a pervasive phenomena in both natural and man-made (including biological) environments. Humans have an innate ability to perceive and take advantage of symmetry in everyday life, but it is not obvious how to automate this powerful insight on man-made intelligent beings, such as robots. On the surface, symmetry is simple and basic. In essence, the concept of symmetry is much more than a mirror reflection, rather, it can span a continuous spectrum of multi-dimensional spaces. In basic sciences, the understanding of symmetry played a profound role in several important discoveries including: relativity theory (the symmetry of time and space); human DNA structure (double helix); the quasi-crystals and their mathematical counterpart penrose tiles. We argue that reasoning about symmetry can likewise play a crucial part in the advance of artificial/machine intelligence.

A computational model for symmetry is especially pertinent to robotics, computer vision and machine intelligence in general, because in these fields we are studying how a man-made intelligent being can perceive and interact with the chaotic real world in the most effective way. Recognition of symmetries is the first step towards capturing the essential structure of a real world problem, and minimizing redundancy which can often lead to drastic reductions in computation. One fundamental limitation of computers is their finite representational power. One simple floating point error can destroy any perfect symmetry. One's ability to tolerate departure from perfect symmetry reflects one's level of sophistication in perception, which need to be built into the development of machine/artificial intelligence.

While a compelling mathematical theory of symmetry has existed for more than a century, very few computational tools prevail in recognizing and taking advantage of real world symmetry. One cause of this shortage is the discrepancy between the ideal algebraic formulation of symmetry, namely group theory, and the instantiation of symmetry in the noisy physical world. We have developed a computational framework that can effectively treat real world symmetry as *statistical departure from regularity*. The final goal of this course is to build perceptual systems that can recognize imperfect structural regularity or symmetry, while discriminating subtle pattern differences.

PLAN: The course will follow a text book on "Symmetry Group Applications" by Dr. Liu and state-of-the-art research papers. The course will be in the format of instructor lectures, student presentations, projects and term papers. Guest lectures (by speakers in and out of CMU) will expose students to applications in other domains (e.g. architecture, material science). Expected number of students: 15 to 25. Expected students: graduate or senior undergraduate. Total of 12 credits. Meeting once per week for a 3 hour-session.

Regularity and Symmetry

What is regularity?
What is symmetry?
Why do we care about symmetry?
How is symmetry related to group theory?

(1) the spectrum of symmetry from regular to stochastic

near-regular texture synthesis: symmetry as a double-sided sward

(2) the formal concept of symmetry and symmetry groups

(3) computational challenges of computational symmetry

Symmetry Groups rising from real world problems

Introduction to some basic concepts in group theory: definition of a group, subgroup, different types of (sub)groups, discrete, continuous, finite, infinitely countable, subgroup hierarchies, transformation groups, matrix representations with concrete examples from

- robotics (surface contact and relative motion between 3D solids),
- computer vision (periodic pattern perception),
- papercut-art form
- biomedical structures/images (the bilateral symmetry of human anatomy).

Coding theory is the study of the properties of codes and their respective fitness for specific applications. Codes are used for data compression, cryptography, error detection and correction, data transmission and data storage. Codes are studied by various scientific disciplines—such as information theory, electrical engineering, mathematics, linguistics, and computer science—for the purpose of designing efficient and reliable data transmission methods. This typically involves the removal of redundancy and the correction or detection of errors in the transmitted data.

There are four types of coding:[1]

1. Data compression (or *source coding*)
2. Error control (or *channel coding*)
3. Cryptographic coding
4. Line coding

Data compression attempts to remove redundancy from the data from a source in order to transmit it more efficiently. For example, Zip data compression makes data files smaller, for purposes such as to reduce Internet traffic. Data compression and error correction may be studied in combination.

Error correction adds extra data bits to make the transmission of data more robust to disturbances present on the transmission channel. The ordinary user may not be aware of many applications using error correction. A typical music CD uses the Reed-Solomon code to correct for scratches and dust. In this application the transmission channel is the CD itself. Cell phones also use coding techniques to correct for the fading and noise of high frequency radio transmission. Data

modems, telephone transmissions, and the NASA Deep Space Network all employ channel coding techniques to get the bits through, for example the turbo code and LDPC codes.

Channel coding

The purpose of channel coding theory is to find codes which transmit quickly, contain many valid code words and can correct or at least detect many errors. While not mutually exclusive, performance in these areas is a trade off. So, different codes are optimal for different applications. The needed properties of this code mainly depend on the probability of errors happening during transmission. In a typical CD, the impairment is mainly dust or scratches.

CDs use cross-interleaved Reed–Solomon coding to spread the data out over the disk.

Although not a very good code, a simple repeat code can serve as an understandable example. Suppose we take a block of data bits (representing sound) and send it three times. At the receiver we will examine the three repetitions bit by bit and take a majority vote. The twist on this is that we don't merely send the bits in order. We interleave them. The block of data bits is first divided into 4 smaller blocks. Then we cycle through the block and send one bit from the first, then the second, etc. This is done three times to spread the data out over the surface of the disk. In the context of the simple repeat code, this may not appear effective. However, there are more powerful codes known which are very effective at correcting the "burst" error of a scratch or a dust spot when this interleaving technique is used.

Other codes are more appropriate for different applications. Deep space communications are limited by the thermal noise of the receiver which is more of a continuous nature than a bursty nature. Likewise, narrowband modems are limited by the noise, present in the telephone network and also modeled better as a continuous disturbance.Cell phones are subject to rapid fading. The high frequencies used can cause rapid fading of the signal even if the receiver is moved a few inches. Again there are a class of channel codes that are designed to combat fading

Convolutional codes

The idea behind a convolutional code is to make every codeword symbol be the weighted sum of the various input message symbols. This is like convolution used in LTI systems to find the output of a system, when you know the input and impulse response.

So we generally find the output of the system convolutional encoder, which is the convolution of the input bit, against the states of the convolution encoder, registers.

Fundamentally, convolutional codes do not offer more protection against noise than an equivalent block code. In many cases, they generally offer greater simplicity of implementation over a block code of equal power. The encoder is usually a simple circuit which has state memory and some feedback logic, normally XOR gates. The decoder can be implemented in software or firmware.

The Viterbi algorithm is the optimum algorithm used to decode convolutional codes. There are simplifications to reduce the computational load. They rely on searching only the most likely

paths. Although not optimum, they have generally been found to give good results in low noise environments.

Convolutional codes are used in voiceband modems (V.32, V.17, V.34) and in GSM mobile phones, as well as satellite and military communication devices.

Cryptographic coding

Cryptography or cryptographic coding is the practice and study of techniques for secure communication in the presence of third parties (called adversaries).More generally, it is about constructing and analyzing protocols that block adversaries;various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiationare central to modern cryptography. Modern cryptography exists at the intersection of the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

Cryptography prior to the modern age was effectively synonymous with *encryption*, the conversion of information from a readable state to apparent nonsense. The originator of an encrypted message shared the decoding technique needed to recover the original information only with intended recipients, thereby precluding unwanted persons from doing the same. Since World War I and the advent of the computer, the methods used to carry out cryptology have become increasingly complex and its application more widespread.

Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system, but it is infeasible to do so by any known practical means. These schemes are therefore termed computationally secure; theoretical advances, e.g., improvements in integer factorization algorithms, and faster computing technology require these solutions to be continually adapted. There exist information-theoretically secure schemes that provably cannot be broken even with unlimited computing power—an example is the one-time pad—but these schemes are more difficult to implement than the best theoretically breakable but computationally secure mechanisms.