ECEN 604 - Channel Coding for Communication
Lectures by Graduate Students

Narayanan Rengaswamy
Department of Electrical and Computer Engineering
Texas A&M University, College Station, TX 77843

October 2015

Lecture 2

1 Abstract Algebra for Engineers

Our goal is defined clearly: to design efficient codes for mitigating the noise introduced by various channels. To convey information with the least amount of error, we should ideally work with real numbers. However, we know that memory, hardware and other practical constraints prevent us from storing and processing real numbers, and hence we resort to digitization of data. More precisely, we store data as a stream of symbols picked from a finite-sized alphabet. Once we frame such an alphabet, we also need to define the interactions between the symbols in the alphabet. If it were a human language, we would call these definitions, comprehensively, as **syntax and semantics**. For us, we need to define the arithmetic that characterizes the interactions between symbols from our alphabet.

Abstract algebra is the class of mathematics that generalizes operations on mathematical objects, focuses on their fundamental properties and characterizes their interactions. The "abstractness" comes from the fact that properties are abstracted from the object. More generally, these objects need not be numbers or other commonly known mathematical notions, necessarily, and can be any abstract object, for e.g. permutations, elements of the periodic table, as long as there are some useful operations that can be defined on them. Depending upon the objects we choose, the theory allows us to define operations relevant to them.

Let us look at the set of real numbers $\mathbb{R}$, a well-known alphabet. $+, -, \times, \div$ are some of the operations defined on $\mathbb{R}$ or a subset $S \subset \mathbb{R}$ of it. In abstract algebra, this set S combined with the operations defined on it is called a field if it satisfies some properties that we will define subsequently. $\mathbb{R}$ with the above mentioned operations is a field. Similarly, the set of complex numbers, $\mathbb{C}$, along with its operations forms a field. As mentioned before, we are interested more in finite fields, i.e. fields with a finite number of objects. The famous set

$\mathbb{F}_2 \triangleq \{0,1\}$ with the operations $(+,-,\times,\div)$ mod 2 is the binary field. As another example, the set $\mathbb{F}_3 \triangleq \{0,1,2\}$ with the operations $(+,-,\times,\div)$ mod 3 is also a finite field.

In this section, we will look at essential concepts of abstract algebra, namely groups and fields. Subsequently, we will use these concepts to understand binary linear codes. Once we familiarize the use of these abstract concepts in the realm of binary codes, we will revisit them in more detail to build up towards the more general construction of non-binary linear codes. A good reference for these concepts is [1, Chapter 2].

1.1 Groups

Definition 1. A **binary operation** $*$ on a set S is a mapping from $(x,y) \in S \times S$ to $z \in S$ denoted by $z = x * y$.

A binary operation $*$ defined on a set S:

(i) is **associative** if $x*(y*z) = (x*y)*z \ \forall\ x,y,z \in S$;

(ii) is **commutative** if $x*y = y*x \ \forall\ x,y \in S$;

(iii) **has identity** if $\exists\ e \in S$ s.t. $x*e = e*x = x$;

(iv) is **invertible** if $\forall\ x \in S\ \exists\ y \in S$ s.t. $x*y = y*x = e$.

It is important to note that a given binary operation need not satisfy all of the above properties.

Definition 2. A **group** $G = (S,*)$ is a set S with a binary operation $*$ defined on it such that

(i) the group is **closed** under that binary operation, i.e. $x*y \in S \ \forall\ x,y \in S$;

(ii) the operation is **associative** ;

(iii) the operation has an **identity** ;

(iv) the operation is **invertible**.

It is important to note that a group does not have to be commutative always, i.e. the binary operation need not be necessarily commutative. If the group is commutative, then it is called an **Abelian group**. A few of examples of groups are:

1. $G = (S,*)$ where $S = \{e,a,b,c\}$ with $*$ defined as below:

$*$	$\boldsymbol{e}$	$\boldsymbol{a}$	$\boldsymbol{b}$	$\boldsymbol{c}$
$\boldsymbol{e}$	e	a	b	c
$\boldsymbol{a}$	a	b	c	e
$\boldsymbol{b}$	b	c	e	a
$\boldsymbol{c}$	c	e	a	b

This group is isomorphic to $G = (\{0, 1, 2, 3\}, + \bmod 4)$.

2. A permutation p on a set S is a bijective function on the set A s.t. $p : A \to A$. Let $A = \{1, 2, 3, 4\}$ and p be defined as follows:

$$p = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{bmatrix}.$$

This means that $1 \to 3, 2 \to 4, 3 \to 1, 4 \to 2$.

Another way to think of p is as an operator in prefix notation such that

$$p \circ 1 = 3, \quad p \circ 2 = 4.$$

We know that there are $n!$ permutations on n elements. If we pick another permutation p_2, then we can define $p_2 \circ p_1$ as the composition of the two permutations, which first applies p_1 and then applies p_2 on a given series of n elements. The result will be another permutation on the n elements. Since the composition of permutations is associative, it can be shown that the set of all permutations on n elements, $S = \{p_1, p_2, \ldots, p_{n!}\}$ is a group under the operation $* = \circ$ defined as the composition of permutations. For details on this example, see [1, Example 2.10].

3. Integers under addition.

4. Matrices of real numbers $\mathbb{R}^{\mathrm{d} \times \mathrm{d}}$ with determinant 1 under multiplication. Note that this is not an Abelian group.

5. Polynomials under addition.

We will use the notation G to denote both the group and its set.

Definition 3. The **order** of a group G is the number of elements in it.

Group Notations

1. **Additive groups**: represented as $G = (S, +)$ and $\underbrace{g + g + \cdots + g}_{k \text{ times}} = kg$. $(-g)$ represents the additive inverse. The additive identity e is represented by 0.

2. **Multiplicative groups**: represented as $G = (S, \times)$ and $\underbrace{g \times g \times \cdots \times g}_{k \text{ times}} = g^k$. g^{-1} represents the multiplicative inverse. The multiplicative identity e is represented by 1.

Definition 4. The **order** of an element $g \in G$ is the smallest positive integer k s.t. $\underbrace{g * g * \cdots * g}_{k \text{ times}} = e$. It is represented as $\operatorname{ord}(g) = k$. If no such finite k exists, then g has infinite order.

Corollary 5. *In general, for $g \in G$, its inverse in the group is inv(g)* $= \underbrace{g * g * \cdots * g}_{ord(g)-1 \ times}$. *For brevity of notation, we write this as $g^{-1} = g^{ord(g)-1}$ even though the group operation is not necessarily multiplication.*

1.1.1 Subgroups

Definition 6. A **subgroup** of a group $(G, *)$ is a subset $H \subseteq G$ such that H is a group.

Theorem 7. *$H \subseteq G$ is a **subgroup** of $(G, *)$ if*

(i) ***closed**, i.e. $x * y \in H \ \forall \ x, y \in H$, and*

(ii) ***inverse exists**, i.e. $\exists \ x^{-1} \in H \ \forall \ x \in H$ s.t. $x * x^{-1} = e$.*

Proof. Verify all the properties of groups as below:

- H is closed, by *(i)*, and has inverses, by *(ii)*.
- H is associative because G is associative.
- $x * x^{-1} = e \ \forall \ x \in H$ and $x * x^{-1} \in H \Rightarrow e \in H$. Hence, H has an identity. □

Definition 8. The **cyclic subgroup**, of a group $(G, *)$, generated by a $h \in G$ with finite order, ord$(h) < \infty$, is

$$H =< h >= \{h, h * h, h * h * h, \ldots\}.$$

The order (or cardinality) of H is ord$(H) = |H| =$ ord(h).

All cyclic subgroups are commutative and hence are Abelian groups under the operation defined on their original group. In this course, we will mainly deal with cyclic subgroups.

1.1.2 Group operations between sets

Definition 9. Let $(G, *)$ be a group and $S, T \subseteq G$. Then,

1. $x * S \triangleq \{x * s | s \in S\} \ \forall \ x \in G$.
2. $S * y \triangleq \{s * y | s \in S\} \ \forall \ y \in G$.
3. $S * T \triangleq \{s * t | s \in S, t \in T\}$.

Note that the output of each of the above operations is a set of elements from G. These operations now allow us to define cosets of a subgroup H in G. Later, we will see that the last operation is useful to define a binary operation between two cosets.

1.1.3 Cosets

Definition 10. A **coset** of a group G is a shifted subgroup.

Example: Let $G = (\mathbb{Z}, +)$. Then,

$$\begin{aligned} H &= \{\ldots, -6, -4, -2, 0, 2, 4, 6, \ldots\} \text{ is a subgroup and} \\ H + 1 &= \{\ldots, -5, -3, -1, 1, 3, 5, 7, \ldots\} \text{ is a coset of } H \text{ in } G. \end{aligned}$$

Definition 11. Let H be a subgroup of $(G, *)$. Then,

$$\begin{aligned} x * H &= \{x * h | h \in H\} \text{ is a left coset of } H \text{ and} \\ H * y &= \{h * y | h \in H\} \text{ is a right coset of } H. \end{aligned}$$

For commutative groups, $x * H = H * x$, i.e. left coset = right coset.

1.1.4 Coset Table

Let $H = \{h_1 = 1, h_2, h_3, \ldots, h_{|H|}\}$ be a subgroup of $(G, *)$. We have taken the first element as the identity w.l.o.g. (without loss of generality). Then, the left coset table is constructed as follows:

1. Write subgroup elements in the first row.
2. Pick any element not in table, say g.
3. Create the next row as $g * H$.
4. Repeat the previous two steps until all elements in G appear in the table.

1	1	h_1	h_2	h_3	$\cdots$	$h_{\|H\|}$
g_1	g_1	$g_1 * h_1$	$g_1 * h_2$	$g_1 * h_3$	$\cdots$	$g_1 * h_{\|H\|}$
g_2	g_2	$g_2 * h_1$	$g_2 * h_2$	$g_2 * h_3$	$\cdots$	$g_2 * h_{\|H\|}$
g_3						
$\vdots$						

The entries in the first column of the table are called **coset leaders**.

Theorem 12. *Each element in G appears exactly once in the coset table.*

Proof. g_i is unused previously by construction.

(a) Suppose any two entries in the same row are equal. Then,

$$g_i * h_j = g_i * h_k \text{ for } j \neq k.$$

But, left multiplying by g^{-1} implies $h_j = h_k$. This is clearly a contradiction.

(b) Suppose two entries in different rows are equal. Then, w.l.o.g.

$$g_i * h_j = g_k * h_l \text{ for } k < i.$$

Now, right multiply by h_j^{-1}. Then,

$$g_i = g_k * \underbrace{h_l * h_j^{-1}}_{\in H},$$

which implies that g_i is in row k. But, the assumption $i > k$ implies that it will not be chosen for the i^{th} row. Again, we have a contradiction. □

Hence, the two important observations from the coset table are:

1. All distinct cosets are disjoint.
2. $|G| = |H| \cdot (\text{index of } H \text{ in } G) = |H| \cdot (G : H)$, where index implies the number of distinct cosets.

Theorem 13. *(Lagrange's Theorem) Let G be a group of finite order. Then, the order of any subgroup H divides the order of the group, i.e. $|H|$ divides $|G|$ without any remainder.*

Proof. This is evident from the two observations mentioned above. □

Definition 14. The vertical bar $|$ means **divides**. We write $a \mid b$ if a divides b (without a remainder).

So, Lagrange's theorem states that for a finite-order group G and any of its subgroups H, $|H| \mid |G|$. We will use this idea of **coset tables** later to design a **decoding algorithm for linear codes**.

1.1.5 Normal Subgroups

Consider any group G and one of its subgroups H. Let $L = \{x * H | x \in G\}$ be the set of all left cosets and $R = \{H * x | x \in G\}$ be the set of all right cosets of H in G.

Definition 15. A subgroup H of a group G is a **normal subgroup** if $x * H = H * x \ \forall \ x \in G$, i.e. $L = R$.

In the above definition, H can be a subgroup of a non-commutative group too. $L = R$ does not imply that H is commutative. It only means that the left and right cosets are equal, i.e. there can be $h_1 \neq h_2$ such that $x * h_1 = h_2 * x$.

Definition 16. The **quotient group** of a group G with respect to one of its normal subgroups H is represented and defined as the set G **modulo** H,

$$G/H = \{x * H | x \in G\},$$

which is the set of all cosets. The operation $\circledast$ on this set is defined as

$$((xH) \circledast (yH)) \triangleq (x * y)H.$$

This is also called the **factor group** of G w.r.t. H.

The definition comes from the application of the 3rd operation from Section 1.1.2:

$$\begin{aligned}(x * H) \circledast (y * H) &= x * (H * y) * H \\ &= x * (y * H) * H \quad (\because H \text{ is a normal subgroup}) \\ &= (x * y) * H.\end{aligned}$$

Example

Let $G = (\mathbb{Z}, +)$ and $H = \{\ldots, -6, -3, 0, 3, 6, \ldots\}$. The cosets are:

$$\begin{aligned}H &= \{z \in \mathbb{Z} | z \bmod 3 = 0\} \\ H + 1 &= \{z \in \mathbb{Z} | z \bmod 3 = 1\} \\ H + 2 &= \{z \in \mathbb{Z} | z \bmod 3 = 2\}.\end{aligned}$$

Then the factor group is given by $G/H = \{H, H + 1, H + 2\}$ with the following operation table:

$\oplus$	$\boldsymbol{H}$	$\boldsymbol{H+1}$	$\boldsymbol{H+2}$
$\boldsymbol{H}$	H	$H+1$	$H+2$
$\boldsymbol{H+1}$	$H+1$	$H+2$	H
$\boldsymbol{H+2}$	$H+2$	H	$H+1$

If we note carefully, we see that this group is *isomorphic* to (mod 3) addition on integers. But, the interesting part is that we did (mod 3) addition without even introducing an operation called division!

Definition 17. Two sets G_1 and G_2 of identical algebraic structure (e.g. groups, fields) are said to be **isomorphic** if there exists a *bijective* mapping $\phi : (G_1, *) \to (G_2, \diamond)$ between them such that, $\forall\ a, b \in G_1$,

$$\phi(a * b) = \phi(a) \diamond \phi(b).$$

For example, two groups $G_1 = (\{a, b\}, *)$ and $G_2 = (\{0, 1\}, \oplus)$ with the following truth tables are (group) isomorphic.

$*$	$\boldsymbol{a}$	$\boldsymbol{a}$
$\boldsymbol{a}$	a	b
$\boldsymbol{b}$	b	a

$\oplus$	**0**	**1**
0	0	1
1	1	0

A **homomorphism** is also a structure-preserving map between two sets of the same algebraic structure, but with a weaker condition: the mapping need not be bijective, i.e. both sets need not even have the same number of elements.

Definition 18. Two sets G_1 and G_2 of identical algebraic structure (e.g. groups, fields) are said to be **homomorphic** if there exists a mapping $\phi : (G_1, *) \to (G_2, \diamond)$ between them such that, $\forall\ a, b \in G_1$,

$$\phi(a * b) = \phi(a) \diamond \phi(b).$$

2 Fields

Definition 19. A **field** $\mathbb{F}$ is a set S and two binary operations $+$ and $\times$, with additive identity 0 and multiplicative identity 1, defined on it that satisfy the following properties:

(i) $(S, +)$ is an Abelian group.

(ii) $(S^*, \times)$ is an Abelian group, where $S^* = S \setminus \{0\}$.

(iii) The distributive law holds: $(a + b) \times c = (a \times c) + (b \times c)$.

Some examples of fields are:

1. The binary field is defined as $\mathbb{F}_2 = \{0, 1\}$ with operations $(+, \times) \mod 2$.
2. For prime p, the set $\mathbb{F}_p = \{0, 1, \ldots, p - 1\}$ with operations $(+, \times) \mod p$ is a field.

For a prime power $q = p^m$, the field $\mathbb{F}_q$ has a non-trivial construction. These are called as **extension fields** of a base field and we will need to define **rings** in order to construct such fields.

Now, we will focus on the application of groups and fields in the construction of binary error correcting codes. Once we get a good understanding of the connection, we will come back to abstract algebra and define rings and fields more formally. Then, we will be in a position to construct Galois fields and, hence, non-binary linear codes of any required error correcting capability.

References

[1] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms.* John Wiley & Sons, Inc., 2005.