SOME APPLIUCATIONS OF MODULES IN CODING THEORY

Anil Kumar Kashyap

College of Food Technology IGKV Raipur (C G) INDIA

Abstract: As we know coding theory solves the problems of the detection and correction of the error by noise in any channel. In this paper, we explain how this can be done using group theory. Also we will discuss modulus and lastly we will form an ISBN code as well as additional information in its specification.

Key words: Group, Module, Subgroup, Coset, ISBN code.

1. Introduction:

Group theory emerged in the early 19th century, initially driven by the study of solutions to polynomial equations. Its roots can be traced back to Mathematicians like Langrage's, Ruffini, and Galois, with Galois being credited with establishing a formal link between group theory and field theory. The Field theory has since expanded, finding applications in various areas of Mathematics, Physics, Chemistry, Computer Science and Electronics Engineering etc. Now, a group can be defined as follows.

A group $(G, .)$ is a set G along with a binary operation $: G \times G \to G$, such that

(i) $(a.b).c = a.(b.c)\ \forall\ a, b, c \in G$

(ii) There exist an Identity element $e \in G$ such that $a.e = e.a = a\ \forall a \in G$ (this Identity element is unique)

(iii) $\forall a \in G, \exists$ an element $\mathrm{b} \in \mathrm{G}$ such that $ab = ba = e$.

A group $(G, .)$ in which $a\,.\,b = b\,.\,a$ for all $a, b\ \in G$ is called an abelian group.

1.1. Theorem: The Identity element of a group is unique.

Proof: is oblivious.

1.2. Theorem: Let G be a group. Then the Inverse of an element of the group is unique.

Proof: is oblivious.

1.3. Definition: Let G be a group. A subset H of G is a subgroup of G if H is a non empty and it is closed under product and inverse. That is, $, b\ \in H \Longrightarrow a^{-1} \in H$ and $\mathrm{a}b \in H$ .If H is a subgroup of G. We write $H \leq G$.

Note: If $H \leq G$, the Identity of G belongs to H as well.

1.4. Definition: Let G be a group and $a \in G$, the smallest positive integer n such that $a^n = 1$ is called the order of a.

1.5. Definition: Let H be a subgroup of a group. For any $a \in G$, the set $aH = \{ah | h \in H\}$ is called a left coset or just coset. An element of a coset is called a representative of the coset.

1.6. Theorem: Let N be a subgroup of a group G. For any $a \in G$. The set of left cosets of H in G partition G. furthermore, for all $u, v \in G, uN = vN \Leftrightarrow v^{-1}u \in N$.

Proof: First of all, as $N \leq G, 1 \in N$. Thus $g \in gN$ for all $g \in G$,that is, $G = \bigcup_{g \in G} gN$. To show the distinct left cosets have been empty intersection, let $uN \cap vN \neq \emptyset$ for some $u, v \in G$. We must show that $uN = vN$. Let $x \in uN \cap vN$. Then $x = un = vm$ for some $n, m \in N$. This gives $u = u(mn^{-1})$. For any $t \in N, ut = v(mn^{-1}t) \in vN$ as $mn^{-1}t \in N$. Thus $\subseteq vN$. Similarly we get $vN \subseteq uN$. Therefore, $uN = vN$ if they have nonempty intersection and we get that set of left cosets partition G.

By the first part of this theorem, we get $uN = vN$ if and only if $u \in vN$, which is equivalent to $v^{-1}u \in N$.

1.7. Definition: If H is a normal subgroup of group G, the set of cosets of H in G again form a group by defining $(aH)(bH) = (ab)H$. This multiplication makes sense as H is normal. This group is called Quotient group and is denoted by $\frac{G}{H}$.

1.8. Theorem: (Langrage's theorem). If H is a subgroup of a finite group G. $|H|$ divides $|G|$ and the number of left cosets of H in G is $\frac{|G|}{|H|}$.

Proof: Let $|H| = n$ and the number of left cosets of H be k. As the set of left cosets partitions G, by the map $F: H \to gH$ defined by $h \to gh$ is a surjection from H to the left coset gH. Further , F is injective as $gh_1 = gh_2 \Longrightarrow h_1 = h_2$. .This proves $|gH| = |H| = n$. Since G is partitioned into k subsets each of cardinality $n, |G| = kn$. Thus $k = \frac{|G|}{n} = \frac{|G|}{|H|}$.

2. Modulus: In group theory the concept of "Modulus" refers to modular arithmetic, which involves working with reminders of the division. Let Z denote the set of integer and for $n \in N$, define $\frac{Z}{nZ} = \{0, 1, 2, 3, \ldots, n-1\}$. We often read $\frac{Z}{nZ}$ as the integer modulo n.

2.1. Definition :(Congruence). $x \equiv y \bmod n$ Means $x - y$ is an integer multiple of n. Equivalently, x and y have the same reminder when divided by n.

2.2. Lemma: For a fixed $n \in N$ and a, a', b, b' integers , we have

(1) $a \equiv b \bmod \text{n} \Leftrightarrow b \equiv a \bmod \text{n}$.
(2) $a \equiv b \bmod \text{n}$ and $b \equiv c \bmod \text{n} \Longrightarrow a \equiv c \bmod \text{n}$.
(3) $a \equiv a' \bmod \text{n}$ and $b \equiv b' \bmod \text{n}$ than $aa' \equiv bb' \bmod \text{n}$ In particular
$a \equiv a' \bmod \text{n} \Longrightarrow ab \equiv a'b' \bmod \text{n}$.

2.3. Definition: (Least Common Multiple). Let $m, n \in N$ the least common multiple of $lcm(m, n) = 1$ is the smallest positive integer divisible by both m and n.

3. Coding theory:

Imagine a situation in which information is being transmitted between two points. The information takes the form of high and low pulses (for example, radio waves or electric currents), which we will label 1 and 0, respectively. As these pulses are sent and received, they are grouped together in blocks of fixed length. The length determines how much information can be contained in one block. If the length is r, there are 2^rdifferent values that a block can have. If the information being sent takes the form of text, each

block might be a character. In that case, the length of a block may be seven, so that $2^7 = 128$ block values can represent letters (both upper and lower case), digits, punctuation, and so on. During the transmission of data, noise can alter the signal so that what is received differs from what is sent. Fig.1 illustrates the problem that can be encountered if information is transmitted between two points.

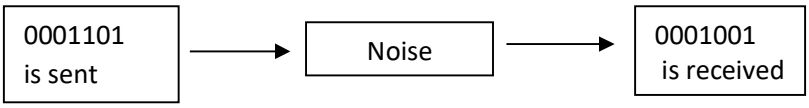

Fig. 1. Noisy transmission

Noise is a fact of life for anyone who tries to transmit information. Fortunately, in most situations, we could expect a high percentage of the pulses that are sent to be received properly. However, when large numbers of pulses are transmitted, there are usually some errors due to noise. For the remainder of the discussion, we will make assumptions about the nature of the noise and the message that we want to send. Henceforth, we will refer to the pulses as bits.

We will assume that our information is being sent along a binary symmetric channel. By this, we mean that any single bit that is transmitted will be received improperly with a certain fixed probability p,independent of the bit value. The magnitude of p is usually quite small. To illustrate the process, we will assume that $p = 0.001$ which, in the real world, would be considered somewhat large. Since $1 - p = 0.999$ we can expect 99.9% of all bits to be properly received.

Suppose that our message consists of 3,000 bits of information, to be sent in blocks of three bits each. Two factors will be considered in evaluating a method of transmission. The first is the probability that the message is received with no errors. The second is the number of bits that will be transmitted in order to send the message. This quantity is called the rate of transmission:

$$\text{Rate} = \frac{\text{Massge length}}{\text{Number of bits transmitted}}$$

As you might expect, as we devise methods to improve the probability of success, the rate will decrease.

Suppose that we ignore the noise and transmit the message without any coding. The probability of success is $(0.999)^{3000} = 0.0497124$. Therefore we only successfully receive the message in a totally correct form less than 5% of the time. The rate of $\frac{3000}{3000} = 1$ certainly doesn't offset this poor probability.

Our strategy for improving our chances of success will be to send an encoded message across the binary symmetric channel. The encoding will be done in such a way that small errors can be identified and corrected. This idea is illustrated in Fig. 2.

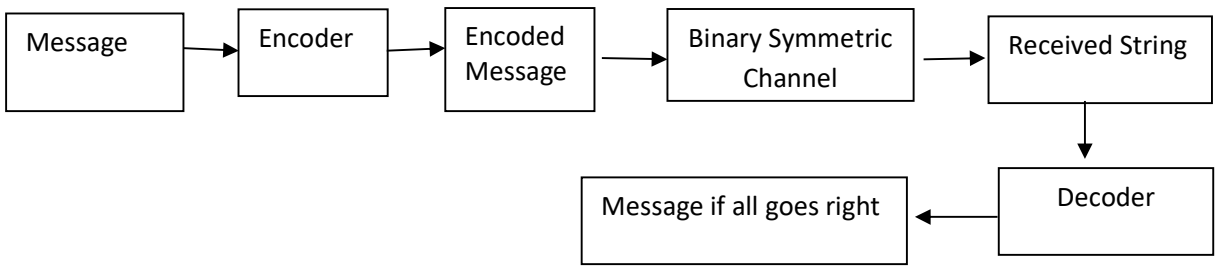

Fig .2. Different stages for message transmission

In our example, the functions that will correspond to our encoding and decoding devices will all be homeomorphisms between Cartesian products of Z_2.

3.1. Error Detection:

Suppose that each block of three bits $a = (a_1, a_2, a_3)$ encoded with the function $e: Z_2^3 \to Z_2^4$, where

$$e(a) = (a_1, a_2, a_3, a_1 +_2 a_2 +_2 a_3)$$

When the encoded block is received, the four bits will probably all be correct (they are correct approximately 99.6% of the time), but the added bit that is sent will make it possible to detect single errors in the block. Note that when $e(a)$ is transmitted, the sum of its components is $a_1 +_2 a_2 +_2 a_3 +_2 (a_1 +_2 a_2 +_2 a_3) = 0$ since $a_i + a_i = 0$ in Z_2.

If any single bit is garbled by noise, the sum of the received bits will be 1. The last bit of $e(a)$ is called the parity bit. A parity error occurs if the sum of the received bits is 1. Since more than one error is unlikely when p is small, a high percentage of all errors can be detected.

At the receiving end, the decoding function acts on the four-bit block $b = (b_1, b_2, b_3, b_4)$with the function $: Z_2^4 \to Z_2^4$, Where $d(b) = (b_1, b_2, b_3, b_1 +_2 b_2 +_2 b_3 +_2 b_4)$

The fourth bit is called the parity-check bit. If no parity error occurs, the first three bits are recorded as part of the message. If a parity error occurs, we will assume that a retransmission of that block can be requested. This request can take the form of automatically having the parity-check bit of $d(b)$sent back to the source. If 1 is received, the previous block is retransmitted; if 0 is received, the next block is sent. This assumption of two-way communication is significant, but it is desirable to make this coding system useful. It is reasonable to expect that the probability of a transmission error in the opposite direction is also 0.001. Without going into the details, we will report that the probability of success is approximately 0.990 and the rate is approximately 3/5. The rate includes the transmission of the parity-check bit to the source.

3.2. Error Correction:

Next, we will consider a coding process that can correct errors at the receiving end so that only one-way communication is needed. Before we begin, recall that every element of $Z_2^n, n \geq 1$, is its own inverse; that is, $-b = b$. Therefore, $a - b = a + b$.

Noisy three-bit message blocks are difficult to transmit because they are so similar to one another. If a and b are in Z_2^3 their difference, $a +_2 b$ can be thought of as a measure of how close they are. If a and b differ in only one bit position, one error can change one into the other. The encoding that we will introduce takes a block $a = (a_1, a_2, a_3)$ and produces a block of length 6 called the code word *of* a. The code words are selected so that they are farther from one another than the messages are. In

fact, each code word will differ from each other code word by at least three bits. As a result, any single error will not push a code word close enough to another code word to cause confusion. Now for the details.

Let $G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$. We call G ia a generator matrix for the code, and let $a = (a_1, a_2, a_3)$ be are our massage.

Define $e: Z_2^3 \to Z_2^6$ by $e(a) = aG = (a_1, a_2, a_3, a_4, a_5, a_6)$

Where

$$a_4 = a_1 \ +_2 \ a_2$$

$$a_5 = a_1 \ +_2 \ a_3$$

$$a_6 = a_1 \ +_2 \ a_4$$

Notice that e is a homomorphism. Also, if a and a are distinct elements of Z_2^3 then $C = a + b$ has at least one coordinate equal to 1. Now consider the difference between $e\ (a)$and $e(b)$.

$$e\ (a) + e(b) = e(a + b)$$

$$= e(c)$$

$$= (d_1, d_2, d_3, d_4, d_5, d_6)$$

Whether c has 1, 2, or 3 ones, $e(c)$must have at least three ones. This can be seen by considering the three cases separately. For example, if c has a single one, two of the parity bits are also 1.

Therefore, $e(a)$and $e(b)$ differ in at least three bits. Now consider the problem of decoding the code words. Imagine that a code word, $e(a)$, is transmitted, and $b = (b_1, b_2, b_3, b_4, b_5, b_6)$is received. At the receiving end, we know the formula for $e(a)$ and if no error has occurred in transmission,

$$b_1 = a_1$$

$$b_1 = a_1$$

$$b_3 = a_3$$

$$b_4 = a_1 \ +_2 \ a_2$$

$$b_5 = a_1 \ +_2 \ a_3$$

$$b_6 = a_1 \ +_2 \ a_4$$

$$\Rightarrow \qquad b_1 \ +_2 \ b_2 \ +_2 \ b_4 = 0$$

$$b_1 \ +_2 \ b_3 \ +_2 \ b_5 = 0$$

$$b_2 \ +_2 \ b_3 \ +_2 \ b_5 = 0$$

The last three equations are called parity-check equations. If any of them are not true, an error has occurred. This error checking can be described in matrix form.

Let

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

P is called parity check matrix for the code. Now define $p: Z_2^6 \to Z_2^3$ by $p(b) = bP$. We call $p(b)$ the syndrome of the received block. For example, $p(0,1,0,1,0,1)$ and $p(1,1,1,1,0,0) = (1,0,0)$.

Note : P is also a homomorphism. If the syndrome of a block is $(0,0,0)$, we can be almost certain that the massage block is (b_1, b_2, b_3).

Next we turn to the method of correcting errors. Despite the fact that there are only eight code words, one for each three-bit block value, the set of possible received blocks is Z_2^6, with 64 elements. Suppose that b is not a code word, but that it differs from a code word by exactly one bit. In other words, it is the result of a single error in transmission. Suppose that w is the code word that b is closest to and that they differ in the first bit. Then $b + w = (1,0,0,0,0,0)$ and

$$p(b) = p(b) + p(w) \text{ since } p(w) = (0,0,0)$$

$$= p(b + w) \text{ since } p \text{ is a homomorphism}$$

$$= p(1,0,0,0,0,0)$$

$$= (1,1,0)$$

Note that we haven't specified b or w only that they differ in the first bit. Therefore, if b is received, there was probably an error in the first bit and $p(b) = (1,1,0)$, the transmitted code word was probably $b + (1,0,0,0,0,0)$ and the message block was $(b_1, +_2 1, b_2, b_3)$. The same analysis can be done if b and w differ in any of the other five bits.

This process can be described in terms of cosets. Let w be the set of code words; that is, $w =$

$e(Z_2^3)$. Since e is a homomorphism, w is a subgroup of Z_2^6. Consider the factor group Z_2^6/w.:

$$\left|Z_2^6/w\right| = \frac{\left|Z_2^6\right|}{|w|} = \frac{64}{8} = 8$$

Suppose that and b_1, b_2 are representatives of the same coset. Then $b_1 = b_2 + w b_1 = b_2 + w$ for some w in W Therefore,

$$p(b_1) = p(b_1) + p(w)$$

Since $p(w) = (0,0,0)$

$$p(b_1) = p(b_1 + w) = p(b_2)$$

and so b_1 and b_2 have the same syndrome.

Theorem: There is a system of distinguished representative of Z_2^6/w Such that each of the six –bit blocks having a single 1 is a distinguished representative of its own coset.

Now we can describe the error-correcting process. First match each of the blocks with a single 1 with its syndrome. In addition, match the identity of W with the syndrome (0,0,0) as in the table below. Since there are eight cosets of W select any representative of the eighth coset to be distinguished. This is the coset with syndrome (1,1,1).

Syndrome	*Error*	*correction*
0 0 0	0 0 0	0 0 0
1 1 0	1 0 0	0 0 0
1 0 1	0 1 0	0 0 0
0 1 1	0 0 1	0 0 0
1 0 0	0 0 0	1 0 0
0 1 0	0 0 0	0 1 0
0 0 1	0 0 0	0 0 1
1 1 1	1 0 0	0 0 1

When block b is received, we need only compute the syndrome $p(b)$, and add to bb the error correction that matches $p(b)$.

4. Applied Coding Theory:

When we hears the word “code” , pictures of computers , zeros and ones and ciphers are most likely the things that comes to the mind. The definition of code is “ a system of signals used to represents letters or numbers in transmitting messages” . Coding is simply defined as converting ordinary language into code. If these definition taken literally, the entire branch of mathematics could be placed under coding. In mathematics, we take ordinary language and convert it into symbols. Other aspects that come to the mind when speaking of coding are secret message and language that only trained personal can translate. Thus one often assumes that the field of coding theory deals with underground spies or secret military language such as Morse code. This is not true and coding theory is frequently confused with cryptography. Cryptography deals with encoding messages so that thery can only be read by the intended receiver.

Coding theory on the other hand focus on assuring that there is no error or redundancy in the delivered massage.

Coding theory applies modular arithmetic profusely throughout its field of practice. One of the most famous problems using modulus in coding theory is the Intentional Standard Book Number (ISBN), found on the back of most books. This ten digit number, if accurate, has the property of

$$10a_1 + 9a_2 + 8a_3 + 7a_4 + 6a_5 + 5a_6 + 4a_7 + 3a_8 + 2a_9 + a_{10} = 0 \bmod 11$$

(where $a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9$ and a_{10} are the ten digits in order from left to right and a_{10} is the check digit).

The check digit is picked specifically so that when it is added to $(10a_1 + 9a_2 + 8a_3 + 7a_4 + 6a_5 + 5a_6 + 4a_7 + 3a_8 + 2a_9)$, the toatal sum will equal $0 \bmod 11$. In this case that the check digit is 10, an X will be written as the final number, for example $0 - 805 - 38703 - \mathrm{X}$. Notice that the integers 1 through 10 are utilized as coefficients since the ISBN is executed using $0 \bmod 11$.

The ISBN number, $0 - 440 - 23697 - 5$, on a widely used calculus book can be checked for accuracy using this scheme,

$$10 * 0 + 9 * 4 + 8 * 4 + 7 * 0 + 6 * 2 + 5 * 3 + 4 * 6 + 3 * 9 + 2 * 7 + 5 = 165 = 0 \bmod 11.$$

The ISBN check scheme is capable to detecting and correcting any single error. Thus if an error was made in the third digit giving $0 - 450 - 23697 - 5$ instead of $0 - 440 - 23697 - 5$, symtem would be able to detect an error since

$$10 * 0 + 9 * 4 + 8 * 5 + 7 * 0 + 6 * 2 + 5 * 3 + 4 * 6 + 3 * 9 + 2 * 7 + 5 = 173 \neq 0 \bmod 11.$$

Because this sum does not equal to $0 \bmod 11$, an error can be detected and the ISBN would be checked or re-entered.

Now we will explain the error detection and correction scheme in ISBN through the following theorem:

4.1. Theorem: The ISBN scheme is able to detect any single digit error.

Proof: Let a_i be any one of the ten digits any the ISBN, when $1 \leq i \leq 10$. When inserting the ISBN into the algorithm, the co-efficient on a_i will be equal to $11 - i$. Let $n = 11 - i$, $1 \leq i \leq 10$. Now let a_i' be an error made in the i[th] digit and let $d = (a_i - a_i')$. In order for the error to go undetected both sums would have to be the congruent modulo 11. That is

$$10a_1 + \cdots + n(a_1) + \cdots + a_{10} = 10a_1 + \cdots + n(a_i') + \cdots + a_{10} \bmod 11$$

$$\Rightarrow n(a_i) - n(a_i') \equiv 0 \bmod 11$$

$$\Rightarrow n(a_i - a_i^{'}) \equiv 0 \bmod 11$$

$$\Rightarrow nd = 0 \bmod 11$$

Thus nd must be divisible by 11. Because 11 is prime, we know that 11 |n or 11|d. since we let $n = 11 - i$ where $1 \leq i \leq 10$, then 11 would be not divide n. We also let $d = (a_i - a_i^{'})$, which implies $1 \leq |d| \leq 10$ (since a_{10} could be X = 10). Then 11 do not divide into d either. Therefore any single digit error will not go undetected.

An extremely noteworthy mathematician, who will remain nameless, once said that after an error is detected in the ISBN scheme , "we can find where the error occurred by reducing the sun S modulo 11, and then computing the additive inverse of S , $-S$, mod 11. The $-S$ th term will be where the error took place. Thus in our previous ISBN on the calculus book, we would compute S=173 mod 11=8 and then obtain the additive inverse of 8 modulo 11 which equal 3 (8+3=0 mod 11). We would then conclude that an error took place in the third position. Now we can correct the error using basic algebra

$$10 * 0 + 9 * 4 + 8 * x + 7 * 0 + 6 * 2 + 5 * 3 + 4 * 6 + 3 * 9 + 2 * 7 + 5 = 165 = 0 \bmod 11$$

$\Rightarrow 8x + 133 \equiv 0 \bmod 11$

$\Rightarrow 8x + 1 \equiv 0 \bmod 11$ since $133 \equiv 1 \bmod 11$

$\Rightarrow 8x \equiv 10 \bmod 11$ since 1 has an additive inverse of $10 \bmod 11$

$\Rightarrow x = 4$ Since 8*4=32≡10 mod 11.

Thus the mathematician would conclude that the third should be 4 . We can check this using the algorithm

$10 * 0 + 9 * 4 + 8 * 4 + 7 * 0 + 6 * 2 + 5 * 3 + 4 * 6 + 3 * 9 + 2 * 7 + 5 = 165 = 0 \bmod 11.$

Using the hypothesis, we would incorrectly assume an error occurred in the six digit, since 6 is the additive inverse of 5 modulo 11. Consequently we could end up changing an accurate digit and in turn create a total error in the ISBN. The most accurate way to correct an error after one has been detected would be to check or retransmit the ISBN.

The ISBN can also detect any by side transposition error. Assume the second and third digoits of the number $0 - 471 - 61884 - 5$ were switched giving $0 - 741 - 61884 - 5$. Using the ISBN system,

$10 * 0 + 9 * 7 + 8 * 4 + 7 * 1 + 6 * 6 + 5 * 1 + 4 * 8 + 3 * 8 + 2 * 4 + 5 = 212 \neq 0 \bmod 11.$

Would be obtained and an error would be detected. A general proof of this can be made as follows.

4.2. Theorem: The ISBN scheme will detect any side by side transposition errors.

Proof Assume $1 \leq i \leq j \leq 10$ and $j = i + 1$. So a_i and a_j will be $(11 - i)$ and $(11 - j)$ respectively. Assume an error occurs switching the order of a_i and a_j so that a_j comes before a_i. If the error were go to undetected, both sums in the algorithm would be congruent modulo 11. Then

$$10a_1 + \cdots + (11 - i)(a_i) + (11 - j)(a_j) + \cdots + a_{10} \equiv 10a_1 + \cdots + (11 - i)(a_j) + (11 - j)(a_i) \ldots + a_{10} \text{mod} 11$$

Thus the difference of the sums would be congruent to $0 \bmod 11$., that is

$$[(11 - i)(a_i) + (11 - j)(a_j)] - [(11 - i)(a_j) + (11 - j)(a_i)] \equiv 0 \bmod 11.$$

$$\Longrightarrow (a_i - a_j)(j - i) \equiv 0 \bmod 11 \text{ when simplified.}$$

Since we know that $(j - i) = 1$ by our assumption, we have $(a_i - a_j) \equiv 0 \bmod 11$. Thus $(a_i - a_j)$ must be divisible by 11. Recall that $1 \leq a_i, a_j \leq 10$ (since a_{10} could be $x = 10$). If we let $d = (a_i - a_j)$ we know that $0 \leq |d| < 11$, which implies $d < 11$. We know that 11 cannot divide d and thus deduce that transposition error would not go undetected.

What if two errors were made in the number $0 - 471 - 40827 - 1$, one in each of the first and seven positions, give

$$10 * 0 + 9 * 4 + 8 * 7 + 7 * 1 + 6 * 4 + 5 * 0 + 4 * 4 + 3 * 2 + 2 * 7 + 1 = 210 = 0 \bmod 11.$$

This system does not catch all two digit errors. The ISBN system is useful in error detection, but not flawless.

References:

[1] Gallian Joseph A., Contemporary Algebra, Haughton Miffin Company Bosten , New York. 2002.

[2] Ling, San, Chaoping Xing, Coding Theory: A First Course, The University of Cambridge. Cambridge United Kingdom 2004.

[3]McEliece, R., *The Theory of Information and Coding*, 2nd edn, Cambridge University Press,Cambridge, UK, 2002.

[4]Pless Vera, Introduction to the Theory of Error Correcting Code, Wiley Inter Science Pub;ication New York,1989.

[5] Raymond Hill, A First Course in Coding Theory, Oxford University Press, 1986.

[6]Strang, G., Linear Algebra and its Applications, 3rd edn, Harcout Brace Jovanovich College Publishers, Orlando, FL, USA, 1988.

[7] Van Lint, J. Introduction to Coding Theory, 3rd edn, Springer Verlag, Berlin, Germany, 1999.

[8] Wozencraft, J.M. and Reiffen, B. , Sequential Decoding, MIT Press, Cambridge, MA, USA. 1961.