

## 1. 代數結構

一個定義於  $A$  的二元運算 " $\circ$ " (Binary Operation) 為一個由  $A \times A$  對應到  $A$  的函數，這裡  $A \times A = \{(x, y) \mid x, y \in A\}$ 。

例如下表就是一個二元運算：

|                   |         |   |   |   |
|-------------------|---------|---|---|---|
|                   | $\circ$ | 1 | 2 | 3 |
| $A = \{1, 2, 3\}$ | 1       | 1 | 3 | 2 |
|                   | 2       | 3 | 2 | 1 |
|                   | 3       | 2 | 1 | 3 |

於是  $\circ((1, 2)) = 3$ ，一般以  $1 \circ 2 = 3$  表示。

所謂的代數結構是指一個集合以及定義於這個集合的二元運算所形成的組合。

**定義 1.1.** 群胚 (Groupoid)

若是  $\circ$  為定義於  $S$  的一個二元運算，則  $\langle S, \circ \rangle$  稱為是一個群胚。

以下是一些  $\langle S, \circ \rangle$  可能有的性質。

(1) 結合律 (Associative Law)

$$\forall a, b, c \in S, a \circ (b \circ c) = (a \circ b) \circ c。$$

(2) 單位元素 (Identity)

若是  $e \in S$  且滿足對於任一元素  $a \in S$ ， $a \circ e = e \circ a$ ，則  $e$  為  $\langle S, \circ \rangle$  的單位元素。

(3) 反元素 (Inverse)

若是  $e$  為  $\langle S, \circ \rangle$  的單位元素，則滿足  $b \circ a = a \circ b = e$  的  $b$  稱為是  $a$  的反元素。

(4) 交換律 (Commutative Law)

$$\forall a, b \in S, a \circ b = b \circ a。$$

(5) 全對稱 (Total Symmetry)

$$\forall a, b \in S, a \circ (a \circ b) = b \text{ and } (a \circ b) \circ b = a。$$

(6) 等冪 (Idempotent)

$$\forall a \in S, a^2 = a \circ a = a。$$

(7) 半對稱 (Semi-symmetric)

$$\forall a, b \in S, a \circ (b \circ a) = b \circ a.$$

(8) 冪等 (Unipotent)

$$\exists c \in S, \text{ s.t. } \forall a \in S, a^2 = c.$$

(9) 拉丁性質 (Latin Property)

$$\forall a, b \in S, a \circ x = b \text{ and } y \circ a = b \text{ 分別在 } S \text{ 中有唯一解。}$$

**定義 1.2.** 半群 (Semigroup)

具有結合律的群胚稱為半群。

**定義 1.3.** 么群 (Monoid)

具有單位元素的半群是么群。

**定義 1.4.** 群 (Group)

如果  $\langle G, \circ \rangle$  是一個么群，而且  $G$  中的每個元素都有反元素，則  $\langle G, \circ \rangle$  為一群。

**定義 1.5.** 交換群 (abelian group)

具有交換律的群稱為是交換群。

**定義 1.6.** 擬似群 (Quasigroup)

具有拉丁性質的群胚是擬似群。

**命題 1.7.** 具有結合律的擬似群為一群。

**命題 1.8.** 具有拉丁性質的半群為一群。

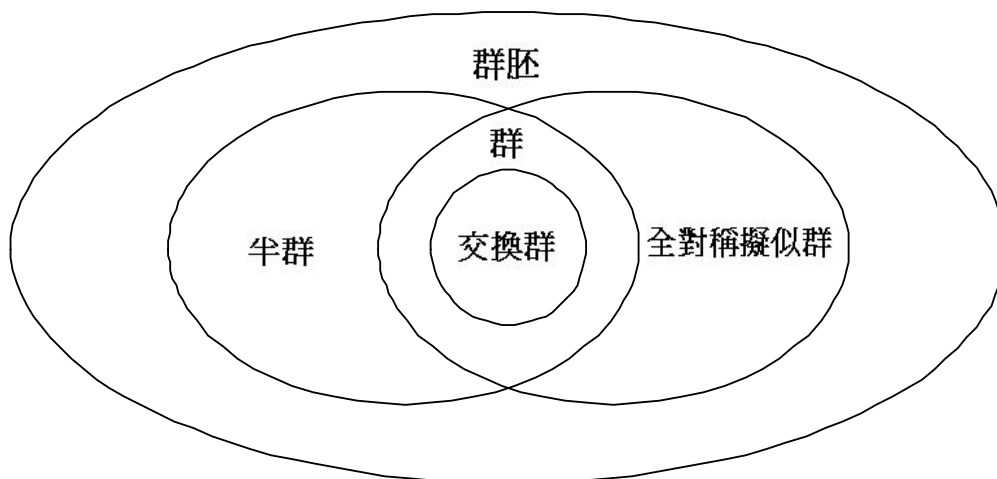
**命題 1.9.** 全對稱的擬似群必為可交換的擬似群。

**命題 1.10.** 全對稱的擬似群必為半對稱的擬似群。

**命題 1.11.** 等冪可交換的擬似群  $\langle Q, \circ \rangle$  存在的充要條件為  $|Q|$  是奇數。

**命題 1.12.** 等冪全對稱擬似群  $\langle Q, \circ \rangle$  存在的充要條件為  $|Q| = 1$  或  $3(mod 6)$ 。

以下我們用一個圖來表示上述一些結構的關係。



**定義 1.13.** 子群 (Subgroup)

令  $H \subseteq G$ ，而且  $\langle H, \circ \rangle$  與  $\langle G, \circ \rangle$  都是群，則  $H$  稱為是  $G$  的子群，一般以  $H \leq G$  表示。

以下的定理最常用來判斷  $H$  是否為  $G$  的子群。

**定理 1.14.** 令  $G$  為一群且  $H \subseteq G$ ，則  $H \leq G$  若且唯若  $H \neq \phi$ ，而且對於  $H$  中的任兩個元素  $x, y$ ， $x \circ y^{-1} \in H$ 。

如果沒有特別的必要性，以下我們將 " $\circ$ " 去掉，用  $xy^{-1}$  來表示  $x \circ y^{-1}$ 。所以，很容易可以證明  $\langle a \rangle = \{a^i | a \in G, i \in \mathbb{Z}\} \leq G$ 。 $\langle a \rangle$  一般稱為是由  $a$  生成 (Generated) 的子群。當  $G = \langle a \rangle$  時， $G$  就是一個循環群 (Cyclic Group)。

(註) 擬似子群的定義與子群相類似，但是定理 1.14 就用不上了。

**定義 1.15.** 餘集 (Coset)

令  $H \leq G$  以及  $a \in G$ ，則  $aH = \{ah | h \in H\}$  為  $H$  的一個左餘集，而  $Ha = \{ha | h \in H\}$  則為  $H$  的一個右餘集。

(註) 餘集定義在群胚上即可。

**命題 1.16.** 令  $H \leq G$ ，則對於任意的  $a, b \in G$ ， $|aH| = |bH|$  而且  $|Ha| = |Hb|$ 。

**命題 1.17.** 令  $H \leq G$  , 則對於任意的  $a, b \in G$  , 不是  $aH = bH$  就是  $aH \cap bH = \phi$  。

**命題 1.18.** 令  $H \leq G$  , 則  $G$  可以寫成互斥左餘集  $(H)$  ( 或右餘集  $(H)$ ) 的聯集。

**定理 1.19.** (Lagrange's Theorem)

令  $H \leq G$  且  $|G| < +\infty$  , 則  $|H||G|$  。

**定理 1.20.** 正規子群 (Normal Subgroup)

令  $H \leq G$  且對於任意  $a \in G$  ,  $aH = Ha$  , 則  $H$  為  $G$  的正規子群 , 以  $H < G$  表示。

上述的條件  $aH = Ha$  可以減弱成  $a^{-1}Ha \subseteq H$  即可。而對於交換群  $G$  , 它的任意子群皆為正規子群。

有了額外的條件 , 我們就可以利用餘集當元素而形成分子群 (Factor Group) 。

**定義 1.21.** (Factor Group 或 Quotient Group)

令  $H < G$  ,  $G/H = \{aH | a \in G\}$  , 且  $(aH) * (bH) = abH$  , 則  $\langle G/H, * \rangle$  形成一群。

(註) 同理 , 可以是  $G/H = \{Ha | a \in G\}$  。

最典型而且最重要的例子是所有整數所成的集合  $Z$  ; 對於加法 " + " ,  $\langle Z, + \rangle$  為一交換群 , 它的子群一定是  $nZ$  的形式 ,  $n$  為任意整數 , 於是分子群  $Z/nZ$  可以定義出來 , 它就是  $Z_n = \{\bar{0}, \bar{1}, \bar{2}, \dots, \overline{n-1}\}$  , 其中  $\bar{i} = i + nZ_n = \{i + nk | k \in Z\}$  是  $nZ$  的一個餘集 , 而在編碼理論中最熱門的群是  $Z_p$  , 其中  $p$  是質數。

**定義 1.22.** 同態函數 (Homomorphism)

令  $\langle G_1, \times \rangle$  與  $\langle G_2, \circ \rangle$  為兩個群。一個函數  $\varphi : G_1 \longrightarrow G_2$  如果滿足  $\forall x, y \in G_1$  ,  $\varphi(x \times y) = \varphi(x) \circ \varphi(y)$  , 則  $\varphi$  稱為是由  $G_1$  映至  $G_2$  的一個同態函數。

(註)

- (1.)  $\varphi$  1-1 ”+” Homomorphism  $\approx$  Monomorphism 。
- (2.)  $\varphi$  onto ”+” Homomorphism  $\approx$  Epimorphism 。
- (3.)  $\varphi$  1-1 及 onto ”+” Homomorphism  $\approx$  Isomorphism ( 同構函數 ) 。

**定義 1.23.** 同態 (Homomorphic)

若是存在一個由  $G_1$  映至  $G_2$  的同態函數，則  $G_1$  與  $G_2$  同態 (Homomorphic to)。(  $G_2$  不一定與  $G_1$  同態 )。

(註) 上述函數如果是同構函數，則  $G_1$  與  $G_2$  同構，以  $G_1 \cong G_2$  表示之。

**定義 1.24.** 自同構函數 (Automorphism)

令  $\langle G, \circ \rangle$  為一個群，則由  $G$  對應到自己的同構函數，也稱為是  $G$  的自同構函數。

由於自同構函數是 1-1 且映成，所以它是一個  $G$  的特殊排列，例如固定每個元素的排列就是一個明顯的自同構函數。

**命題 1.25.** 自同構群 (Automorphism Group)

對於任一個群  $\langle G, \circ \rangle$ ，所有的自同構函數形成一個群，它的二元運算是函數的合成 (Composition)。

接下來我們看具有兩個運算的代數結構。

**定義 1.26.** 環 (Ring)

$\langle R, +, \cdot \rangle$  稱為是一個環，如果下列條件滿足：

- (1)  $\langle R, + \rangle$  為一交換群。
- (2)  $\langle R, \cdot \rangle$  為一半群。
- (3)  $\forall a, b, c \in R, a \cdot (b + c) = a \cdot b + a \cdot c$  ( 分配律 )。

**定義 1.27.** 交換環 (Commutative Ring)

具有乘法交換律的環為交換環。

**定義 1.28.** 單位 (Unit)

在一個環中，乘法單位元素一般以 ”1” 表示，也稱為是環的單元。

(註) 環的定義分成兩派，一個加入單元，一個則沒有單元。為了方便討論，我們採用沒有單元的定義方式。

除了  $\langle \mathbb{Z}, +, \cdot \rangle$  這個出名的交換環， $R[x]$ ，所有實係數多項式所形成的集合，也是非常好的例子；在這裡多項式的加法與乘法是一般的運算方式。

**定義 1.29. 子環 (SubRing)**

如果  $\langle R, +, \cdot \rangle$  是一個環， $R' \subseteq R$ ，而且  $\langle R', +, \cdot \rangle$  也是一個環，則  $R'$  稱為是  $R$  的子環。

**命題 1.30.** 令  $R$  為一個具有單元的交換環，則對於任意一個  $a \in R$ ， $\langle a \rangle = \{an \mid n \in R\}$  形成  $R$  的一個子環。

**定義 1.31. 理想 (Ideal)**

令  $R$  為一環， $I$  為  $R$  的一個子環。如果對於  $R$  中的任一個元素  $n$  都滿足  $nI \subseteq I$ ，則  $I$  稱為是  $R$  的左理想 (left Ideal)，滿足  $In \subseteq I$  的子環則為  $R$  的右理想。同時是左理想也是右理想的  $I$  稱為是  $R$  的一個理想。

(註) 在交換環中，左理想也是右理想。

**命題 1.32.** 在一個具有單元的交換環  $R$  中， $\forall a \in R$ ， $\langle a \rangle$  為  $R$  的一個理想。

(註) 上述形式的理想也稱為是主理想 (Principal Ideal)，它是非常重要的一種理想。

現在舉一個例子來說明  $R = R[x]$ ，即實係數多項式所成的集合，所以  $R$  是一個具有單元的交換環。考慮  $n = x^2 + 1$ ， $\langle x^2 + 1 \rangle$  是  $R[x]$  的一個理想。

**定義 1.33. 環的特徵值 (Characteristic)**

令  $R$  為一環。如果存在一個正整數  $n$  使得  $\forall a \in R$ ， $na = 0$ ，則  $n$  稱為是  $R$  的特徵值，以  $ch(R) = n$  表示。如果  $n$  不存在，則令  $ch(R) = 0$ 。

**例.**  $ch(\mathbb{Z}_m) = m$ ， $ch(\mathbb{Z}) = 0$ 。

**定義 1.34. 除法環 (Division Ring)**

令  $R$  為具有單元的環。如果  $R$  中不為 0 的元素都有乘法反元素，則  $R$  稱為是除法環。

**定義 1.35. 體 (Field)**

一個可以交換的除法環稱為體。

**定義 1.36. 有限體 (Finite Field)**

具有有限多個元素的體稱為有限體。

**定理 1.37. 有限體特性**

令  $\langle F, +, \cdot \rangle$  為一個有限體，則  $|F|$  為質數次方數。

最常見的有限體應該是  $Z_p$ ，其中  $p$  為質數。如果要建構一個有限體，它具有  $p^2$  個元素，或是更一般的形式  $p^k$  個元素， $k > 1$ ，則以下的定理是非常重要的。在敘述定理之前先介紹一個定義。

**定義 1.38. 不可分解 (Irreducible)**

一個佈於體  $F$  的多項式，如果除了本身及常數項之外沒有其它佈於  $F$  的因式，則此多項式為不可分解的多項式。

(註)  $F$  的選擇與可不可分解有很大的關聯性，例如，當  $F = R$  時， $x^2 + 1$  是不可分解的多項式，但是當  $F = C$  或  $Z_2$  時， $x^2 + 1$  就可以分解。

**定理 1.39.**  $F[x]/\langle f(x) \rangle$  為一體的充要條件是  $f(x)$  不可分解 (佈於  $F$ )。

於是，只要令  $F = Z_p$ ，並且找到一個佈於  $Z_p$  的不可分解之  $k$  次多項式，所構出來的體就恰好有  $p^k$  個元素。

為了紀念 Galois 在研究體方面的重大貢獻，具有  $q$  個元素的有限體一般就用  $GF(q)$  來表示，再一次我們示 " $q$  為質數的次方數"。以下我們進一步研究有限體的一些特性。

**定義 1.40. (Primitive Element)**

如果一個體中的所有非零元素都可以寫成一個元素  $\alpha$  的次方，則  $\alpha$  稱為是  $F$  的 primitive 元素。

**定理 1.41.** 令  $F$  為一有限體，則  $\langle F^*, \cdot \rangle$  為一循環群。 $(F^* = F \setminus \{0\})$

這個定理基本上是證明  $F$  中必存在有 primitive 元素。

雖然，我們已建構出有次方個元素的有限體；但是，要說明其它的個數不可能，仍需一些努力。

**定義 1.42. 向量空間 (Vector Space)**

令  $F$  是一個體。一個佈於  $F$  的向量空間 ( $F$ -vector Space) 是指一個交換群  $\langle V, + \rangle$  , 它同時具有以下的 5 個性質 :

- (1)  $\forall a \in F, \alpha \in V, a\alpha \in V$  。 ( 量積 , Scalar Multiplication )
- (2)  $\forall a, b \in F, \alpha \in V, a(b\alpha) = (ab)\alpha$  。
- (3)  $\forall a, b \in F, \alpha \in V, (a + b)\alpha = a\alpha + b\alpha$  。
- (4)  $\forall a \in F, \alpha, \beta \in V, a(\alpha + \beta) = a\alpha + a\beta$  。
- (5)  $1\alpha = \alpha$  。(1 為  $F$  的乘法單位元素)

在上述的定義中,  $V$  的元素稱為是向量, 而  $F$  的元素則稱為純量。

**定義 1.43. 線性組合 (Linear Combination)**

令  $V$  為一個佈於  $F$  的向量空間,  $S = \{\alpha_i | i \in I\}$  為  $V$  中的一組向量所成的集合。如果  $\alpha = \sum_{i \in I} a_i \alpha_i$ , 其中  $a_i \in F$ , 則  $\alpha$  為  $S$  中向量的線性組合。如果  $\left\{ \sum_{i \in I} a_i \alpha_i \mid a_i \in F \right\} = V$ , 則  $S$  中的向量可以展成 (Span)  $V$ , 以  $\langle S \rangle = V$  表示。

**例.**  $R^n$  為一佈於  $R$  的向量空間,  $R^n$  可以由  $e_1, e_2, \dots, e_n$  展成, 其中  $e_i = (0, 0, \dots, 0, 1, 0, \dots, 0)$  (1 在第  $i$  個位置),  $i = 1, 2, \dots, n$ ;  $R^n$  也可以由  $S = \{(1, 0, 0, \dots, 0), (1, 1, 0, 0, \dots, 0), \dots, (1, 1, \dots, 1)\}$  所展成。

**定義 1.44. 線性獨立 (Linear Independence)**

一組  $V$  中的向量  $\{\alpha_i | i \in I\}$  是線性獨立 (佈於  $F$ ) 的集合若且唯若當  $a_i \in F$  且  $\sum_{i \in I} a_i \alpha_i = 0$  時, 對於所有的  $i \in I, a_i = 0$ 。

**定義 1.45. 線性相依 (Linear Dependence)**

不線性獨立 (佈於  $F$ ) 的一組向量就是線性相依 (Dependent)。

**定義 1.46. 基底 (Basis)**

令  $V$  為佈於  $F$  的向量空間。如果  $B = \{\beta_i | i \in I\}$  為一組線性獨立 (佈於  $F$ ) 的向量, 而  $B$  中的向量可以展成  $V$ , 則  $B$  稱為是向量空間  $V$  (佈於  $F$ ) 的一組基底。

**定義 1.47.** 有限維空間 (Finite Dimensional Vector Space)

一個佈於  $F$  的向量空間  $V$  如果能被有限個向量所展成，則此向量空間為有限維的向量空間。

**定理 1.48.** 令  $V$  為佈於  $F$  的有限維空間，則任兩個基底具有相同個數的向量。

由於上述定理在有限維空間中，基底的向量數就被定義為該向量空間的維數 (Dimension)，以  $\dim V$  表示，所以  $\dim R^n = n$ 。

**定理 1.49.** 每一個有限體的元素個數必為質數次方數。

**證明.**

令  $F$  為一個有限體。由於  $1 \in F$  and  $ch(F) < +\infty$ ,  $\therefore ch(F) = p$ ,  $p$  為質數。因此  $F$  必包含一個子體  $GF(p)$ 。令  $\alpha$  為一個 primitive 元素， $m$  為最小的正整數滿足  $\alpha^m = \sum_{i=0}^{m-1} a_i \alpha^i$ ，其中  $a_i \in GF(p)$ 。於是，對於任意的  $n > m$ ， $\alpha^n$  也可以寫成  $1, \alpha, \alpha^2, \dots, \alpha^{m-1}$  的線性組合；也就是說  $F^*$  中的元素均可以用  $1, \alpha, \alpha^2, \dots, \alpha^{m-1}$  的線性組合表示出來。由於這些元素形成一個線性獨立 (佈於  $GF(p)$ ) 的集合，所以  $\{1, \alpha, \alpha^2, \dots, \alpha^{m-1}\}$  可以看成是向量空間  $F$  (佈於  $GF(p)$ ) 的一組基底，於是  $|F| = p^m$ 。 ■

有限體還有一個非常重要的特性。

**定義 1.50.** 同構體 (Isomorphic Fields)

令  $F$  及  $F'$  為兩個體，如果以  $F$  的條件成立，則  $F$  與  $F'$  同構，以  $F \cong F'$  表示。

(註) 存在一個由  $F$  映到  $F'$  的 1-1 映成函數  $\varphi$ ，使得  $\forall \alpha, \beta \in F$ ,  $\varphi(\alpha + \beta) = \varphi(\alpha) + \varphi(\beta)$ ,  $\varphi(\alpha \cdot \beta) = \varphi(\alpha) \cdot \varphi(\beta)$ 。

(註)  $< F^*, \cdot >$  是循環群，其中  $F^* = F \setminus \{0\}$ 。

**證明.**  $F^*$  是有限體的交換群，所以  $F^* \cong Z_{d_1} \times Z_{d_2} \times \dots \times Z_{d_r}$ ，其中  $d_i$  均為質數方。令  $m = l.c.m.\{d_1, d_2, \dots, d_r\}$ ，則  $m \leq d_1 d_2 \dots d_r$ 。再令  $(a_1, a_2, \dots, a_r)$  為  $F^*$  中的一個元素，則它必滿足  $a_i^{d_i} = 1$ ,  $i = 1, 2, \dots, r$ 。於是  $(a_1, a_2, \dots, a_r)^m = (1, 1, \dots, 1)$  為  $F^*$  的單位元素 (乘法)， $\therefore \forall \alpha \in F^*$ ,  $\alpha^m = 1$ ,  $\alpha$  為  $x^m - 1$  的零位； $\alpha$  有  $d_1 d_2 \dots d_r$  個，所以， $m \geq d_1 d_2 \dots d_r$ ，則  $m = d_1 d_2 \dots d_r$ ，因此， $(d_i, d_j) = 1$ ，這表示  $(a_1, a_2, \dots, a_r)$  為 generator。 ■

**定理 1.51.** 任兩個具有相同元素的有限體均同構。

**證明.** (略)

由上述定理可以推得有限體的唯一性，至於對於任意的  $p^m$  都可以建構一個有限體出來就須要先證明：

**定理 1.52.** 對於任意的正整數  $m$  都存在一個 ( 佈於  $\mathbb{Z}_p$  ) 不可分解的  $m$  次多項式。

詳細證明請參考 E.R.Berlekamp 所著的 Algebraic Coding Theory。

## 2. 基本概念

編碼理論的產生自然是來自訊息的傳遞。如何快速而且正確地將一個訊息 (Message) 經過一個管道 (Channel) 傳至對方是訊息傳遞的主要工作。爲了要快，在硬體設備固定的情況之下，代表一個訊息的字串當然是越短越好；然而，在可能產生錯誤的情況之下，越短的訊息就越容易被誤解。

譬如，我們分別用 00,01,10,11 來代表甲，乙，丙，丁四個字；在傳遞時依次傳出 1 或 0。顯然，在絕不產生錯誤的情況下，一切順利；不過，由於傳遞時可能會受到干擾 (Noise)，如聲音，大氣層，雷電，雜訊等，而造成錯誤，例如 01 傳成 11，則乙就被誤以爲是丁，而造成嚴重錯誤。現在，改用 000,011,101,110 來代表甲，乙，丙，丁四個字，那麼情況可能會好轉，當 011 被傳成 010 時，由於四個字中沒有一個是用 010 代表，所以我們知道：一定傳錯了！然而，錯在什麼地方呢？這又是一件頭痛的事了，如何才能知道有錯，而且可以改回來正確的訊息，是我們這門課主要研究的內容。

也許我們會想像上述的工作是不可能完成的任務 (Mission Impossible)；以下是一個例子：令 000000,010101,101010,111111 分別代表甲，乙，丙，丁四個字，於是當錯誤（一個）產生時，如 110101 是收到的字，這個時候，我們自然會判斷正確答案應該是 010101。爲何不是 111111 或其它呢？這當然也需要理由來支持它，至少，我們相信把錯誤找出來是可能的事。至於要如何設計才會比較有效率，就得仰賴各位的智慧了。

爲了方便解釋，也是因爲實用，接下來我們所傳遞的字母 (Digit, bit) 只用 0 與 1。令  $K = GF(2)$ ，所以  $K^n = \{(x_1, x_2, \dots, x_n) \mid x_i \in K, i = 1, 2, \dots, n\}$ 。爲了方便起見， $K^n$  中的向量  $(x_1, x_2, \dots, x_n)$  可以直接寫成  $x_1x_2 \cdots x_n$ 。

### 定義 2.1. 數碼 (Code)

一個  $K^n$  的子集合就稱爲是長度爲  $n$  的數碼 (Code)。

**例 1.**  $C = \{1101000, 0110100, 0011010, 0001101, 1000110, 0100011, 1010001\}$  是一個長度爲 7 的數碼。

### 定義 2.2. 碼字 (Codeword)

一個數碼中的向量稱爲是碼字。

(註) 任一個  $K^n$  中的向量也稱為是長度  $n$  的字 (word)。

**定義 2.3.** 距離 (Distance)

令  $\mathbf{x} = x_1x_2x_3 \cdots x_n$  與  $\mathbf{y} = y_1y_2y_3 \cdots y_n$  為  $K^n$  中的兩個字, 則  $\bar{x}$  與  $\bar{y}$  的距離 (Hamming Distance) 定義成  $d(\mathbf{x}, \mathbf{y}) = |\{i | x_i \neq y_i, i = 1, 2, \dots, n\}|$ 。一個數碼  $C$  的距離  $d(C)$  則定義為  $d(C) = \min \{d(\mathbf{x}, \mathbf{y}) | \mathbf{x}, \mathbf{y} \in C \text{ and } \mathbf{x} \neq \mathbf{y}\}$ 。

**命題 2.4.** 距離為一度量 (Metric), 亦即滿足下列三條件:

- (1)  $d(\mathbf{x}, \mathbf{y}) = 0$  若且唯若  $\mathbf{x} = \mathbf{y}$ 。
- (2)  $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}) \forall \mathbf{x}, \mathbf{y} \in K^n$ 。(對稱性)
- (3)  $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y}) \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in K^n$ 。(三角不等式)

**定義 2.5.** 二元碼 (Binary Code)

當  $K = GF(2)$  時  $K^n$  的子集為二元碼。

**定義 2.6.** 線性碼 (Linear Code)

當數碼  $C$  為  $K^n$  的向量子空間時,  $C$  稱為是線性碼。

**定義 2.7.** 加權 (Weight)

$\mathbf{x} \in K^n$ ,  $\mathbf{x}$  的加權為  $wt(\mathbf{x}) = d(\mathbf{x}, \bar{0}) = |\{i | x_i \neq 0, i = 1, 2, \dots, n\}|$ 。

**命題 2.8.**  $d(\mathbf{x}, \mathbf{y}) = wt(\mathbf{x} + \mathbf{y})$ 。

**命題 2.9.** 當  $wt(\mathbf{x}) = wt(\mathbf{y})$ ,  $d(\mathbf{x}, \mathbf{y})$  為一偶數。

**命題 2.10.**  $wt(\mathbf{x}) + wt(\mathbf{y}) \geq wt(\mathbf{x} + \mathbf{y})$ 。

**命題 2.11.** 當  $C$  為線性碼時,  $d(C) = \min \{wt(\mathbf{x}) | \mathbf{x} \in C \text{ and } \mathbf{x} \neq \bar{0}\}$ 。

**證明.** 因為  $\bar{0} \in C$ , 所以  $d(C) \leq \min \{wt(\mathbf{x}) | \mathbf{x} \in C \text{ and } \mathbf{x} \neq \bar{0}\}$ 。另外, 當  $\mathbf{x} \neq \mathbf{y}$  時,  $d(\mathbf{x}, \mathbf{y}) = wt(\mathbf{x} + \mathbf{y}) \geq \min \{wt(\mathbf{x}) | \mathbf{x} \in C \text{ and } \mathbf{x} \neq \bar{0}\}$ ,  $\mathbf{x} + \mathbf{y} \in C$ , 故得證。 ■

**命題 2.12.** 如果  $C$  為  $k$  維的線性碼, 則  $C$  中有  $2^k$  個碼字。

**命題 2.13.** 如果  $C$  為  $k$  維的線性碼，則  $C$  有  $\frac{1}{k!} \prod_{i=0}^{k-1} (2^k - 2^i)$  個不同的基底。

**證明.** 逐一選取互相獨立的  $k$  個向量。 ■

**定義 2.14.** 對偶碼 (Dual Code)

令  $C$  為一長度  $n$  的線性碼，則  $C^\perp = \{\mathbf{u} \in K^n \mid \mathbf{u} \cdot \mathbf{v} = 0, \forall \mathbf{v} \in C\}$  稱為是  $C$  的對偶碼。

對偶碼在編碼理論的研究上與數碼本身具有同等的重要性，如果  $C^\perp = C$ ，則  $C$  稱為是自對偶碼 (Self-dual)。

**命題 2.15.**  $C^\perp$  為一長度  $n$  線性碼，而且當  $\dim(C) = k$  時， $\dim(C^\perp) = n - k$ 。

**問題.**  $C$  的距離與  $C^\perp$  的距離關聯性如何？

**定理 2.16.** (Hamming Bound)

令  $C$  為一長度  $n$ ，距離  $d$  的數碼，則  $|C| \cdot \left[ \sum_{i=0}^{\delta} \binom{n}{i} \right] \leq 2^n$ ，其中  $\delta = \lfloor \frac{n-1}{2} \rfloor$ 。

**定義 2.17.** 完美碼 (Perfect Code)

若是  $d$  為奇數，使得上述定理不等式成為等式的  $C$ ，稱為是完美碼。

**例.**  $n = 7, d = 3$  (完美碼的例子)

$C = \{1101000, 0110100, 0011010, 0001101, 1000110, 0100011, 1010001, 0010111, 1001011, 1100101, 1110010, 0111001, 1011100, 0101110, 0000000, 1111111\}$ 。

$$|C| = 16, 16 \times \left[ \binom{7}{0} + \binom{7}{1} \right] = 2^7。$$

除了上述用組合的方式建構出來  $(7, 4, 3)$ -完美碼，也可以利用聯立方程式的概念求得。

$$\text{令 } C = \left\{ \mathbf{v} \in K^7 \mid \mathbf{v}H = \bar{0}, H^t = \begin{bmatrix} 0011101 \\ 0100111 \\ 1001011 \end{bmatrix} \right\}。$$

則  $C$  為一個完美碼 (?)。其實，這個  $C$  與上面例子中的數碼是同構的數碼，它的理由並不難理解 (?)。

另外，當  $n = 23$ ， $d = 7$  時也可以建構完美碼，這個數碼將在下一節介紹。除了上述兩個二元完美碼之外，如果取  $K = GF(3)$ ， $n = 11$ ， $d = 5$ ，也可以產生完美碼。

**定義 2.18.** 長度  $n$ ，維數  $k$  以及距離為  $d$  的線性碼以  $(n, k, d)$ -碼表示，如果  $d$  不考慮，則稱為  $(n, k)$ -碼。

**定義 2.19.** 生成矩陣 (Generator Matrix)

若  $C$  為一個  $(n, k)$ -碼，則以  $C$  的基底為列向量所寫成的  $k \times n$  矩陣稱為是  $C$  的生成矩陣。

例.  $C = \{0000, 1110, 0111, 1001\}$  為  $(4, 2)$ -碼， $G = \begin{bmatrix} 1110 \\ 0111 \end{bmatrix}$  為一生成矩陣。  
 $G' = \begin{bmatrix} 1001 \\ 0111 \end{bmatrix}$  也是。

**命題 2.20.** 如果  $C$  為一個  $(n, k)$ -碼，則必存在一個生成矩陣，它的形式為  $[I_k | A]_{k \times n}$ ，其中  $I_k$  為  $k$  階單位矩陣。

(註) 上述的形式又稱為是標準型的生成矩陣 (Standard)。

有了生成矩陣，編碼的工作就可以順利進行。因為一個  $(n, k)$ -碼有  $2^k$  個碼字，也就代表著要傳遞的訊息可以有  $2^k$  個，當然可以少，但是不能多。現在，用維度為  $k$  的字來代表訊息，即令  $\mathbf{v} \in K^k$ ，於是， $\mathbf{v}G$  代表矩陣中列向量的線性組合，當然就是一個碼字了，這個工作就叫做編碼 (Encode)，令  $\mathbf{u} = \mathbf{v}G$ 。如果  $G$  為標準的生成矩陣，則  $\mathbf{u}$  的前面  $k$  個位置恰好形成向量  $\mathbf{v}$ 。

例.  $(1, 1) \begin{bmatrix} 1001 \\ 0111 \end{bmatrix} = 1110$

由上面的描述可以發現，長度為  $n$ ，維數是  $k$  的線性碼，雖然傳送訊息的長度皆為  $n$ ，但是真正代表訊息的只有  $k$  個數字而已，因此，我們就定義  $k/n$  為該數碼的訊息率 (Information Rate)。

除了用生成矩陣來看一個  $(n, k)$ -碼之外，我們也可以用對偶碼的生成矩陣來看這個碼的碼字。

**命題 2.21.** 令  $H^t$  為  $C^\perp$  的生成矩陣,  $G$  為  $C$  的生成矩陣, 則  $GH = [0]$ 。

**證明.** 因為  $H^t$  的每一列與  $G$  的每一列都垂直, 故得證。 ■

**定義 2.22.** 核對矩陣 (Parity Check Matrix)

當  $H^t$  為  $C^\perp$  的生成矩陣時,  $H$  稱為是  $C$  的核對矩陣。

(註) 如果  $H$  為  $C$  的核對矩陣, 則對於  $C$  中的任一碼字  $\mathbf{v}$ ,  $\mathbf{v}H = \bar{0}$ 。反過來, 我們也可以先給一個  $n \times (n-k)$  的矩陣  $H$ ,  $H$  的行向量形成一組線性獨立的集合, 則  $C' = \{\mathbf{v} | \mathbf{v}H = \bar{0}\}$  為一  $(n, k)$ -碼。

**命題 2.23.** 令  $H$  為  $C$  的核對矩陣, 而且  $H$  中任意  $t-1$  列均線性獨立, 則  $d(C) \geq t$ 。

**證明.**  $C$  中不為 0 的碼字加權至少為  $t$ 。 ■

**定理 2.24.** 一個  $(n, k, d)$ -碼必須滿足  $d \leq n - k + 1$ 。

**證明.** 令  $H$  為  $(n, k, d)$ -碼  $C$  的核對矩陣, 則  $H$  為  $n \times (n-k)$  的矩陣。所以互相獨立的列數至多為  $n-k$ 。然而,  $d(C) = d$ , 所以  $H$  中的  $d-1$  列必然形成一個線性獨立的集合, 於是  $n-k \geq d-1$ , 即  $d \leq n-k+1$ 。 ■

**定義 2.25.** 最大距離分隔碼 (Maximum Distance Separable Code, MDS Code)

滿足  $d = n - k + 1$  的  $(n, k, d)$ -碼稱為是最大距離分隔碼。

**定理 2.26.** 令  $C$  為一  $(n, k, d)$ -碼, 則下列敘述等價。

- (1)  $d = n - k + 1$ 。
- (2) 在  $C$  的核對矩陣中, 任意  $n-k$  列均形成線性獨立的集合。
- (3) 在  $C$  的生成矩陣中, 任意  $k$  行均形成線性獨立的集合。

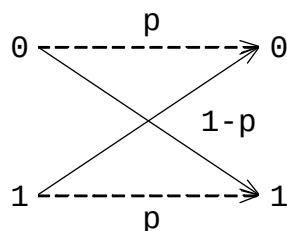
前面介紹的生成矩陣在編碼上扮演重要的角色, 而核對矩陣則在矯碼上提供它的作用, 最簡單的現象是當收到的字  $\mathbf{w}$ , 只要它造成  $\mathbf{w}H \neq \bar{0}$ , 就知道錯誤已經產生了。接下來, 我們討論更正確的概念。

**定義 2.27.** 二元管道 (Binary Symmetric Channel, BSC)

傳遞 0 或 1 具有相同正確性的管道叫做對稱管道。

**定義 2.28.** 可靠度 (Reliability of BSC)

傳遞一個字母會正確地傳達的機率叫做可靠度。



所以，傳錯的機率為  $1-p$ ，在此定義這個機率不會隨著所傳的是 0 或是 1 而有所不同；同時，我們可以令  $1 > p > \frac{1}{2}$ 。至於當  $p = \frac{1}{2}$  時，誰也無法在傳送上做任何修正。

在沒有討論如何解碼之前，先看看產生錯誤的影響力有多大！假設可靠度是  $1 - 10^{-8}$ ，同時傳輸的速度為每秒  $10^7$  個字母，現在要傳長度為 11 的所有字，而傳錯的機率為

$$11p^{10}(1-p) + \binom{11}{2}p^9(1-p)^2 + \cdots + \binom{11}{11}(1-p)^{11} \approx 11p^{10}(1-p) \approx 11 \times 10^{-8}$$

所以大約每 10 秒會傳錯一個字而無法偵測出來，於是每天就會有 8640 個字傳錯，顯然情況不甚理想！

現在，把要傳輸的  $K^{11}$  中的字，每個字在最後加上一個字母，使得每個字都有偶數個 1，亦即加權為偶數，如此一來，產生一個字母錯誤時就可以被發現 (?)，而不至於造成錯誤，因此，傳錯字的機率不會大於

$$\binom{12}{2}p^{10}(1-p)^2 + \binom{12}{4}p^8(1-p)^4 + \cdots + \binom{12}{12}(1-p)^{12} \approx \binom{12}{2}p^{10}(1-p)^2 \approx 66 \times 10^{-16}$$

於是，照上述的計算方法，大約 2000 秒才會出現傳錯一個字的現象，這是非常大的改善；當然傳一個訊息所花的力氣是增加了一點，由 11 到 12，效果卻是驚人。然而有些狀況下，例如太空照相，光只是測出有錯誤並沒有太大的幫助，因為沒有可能重送一次；有能力更正才是重要。因此，毫無疑問的，代表一個訊息的碼字又要增長了！

從前面的描述，我們不難得到一個概念，由於傳錯一個字母的機率，也就是可靠度通常十分接近 1，所以在傳輸一個長度為  $n$  的字時，錯越多個的機率越小。所以，如果令  $\phi_p(\mathbf{v}, \mathbf{w})$  代表傳送  $\mathbf{v}$  而收到  $\mathbf{w}$  的機率，以下的性質成立。

**命題 2.29.**  $\phi_p(\mathbf{v}_1, \mathbf{w}) \leq \phi_p(\mathbf{v}_2, \mathbf{w})$  iff  $d(\mathbf{v}_1, \mathbf{w}) \geq d(\mathbf{v}_2, \mathbf{w})$ 。

**證明.** 令  $d_1 = d(\mathbf{v}_1, \mathbf{w})$ ,  $d_2 = d(\mathbf{v}_2, \mathbf{w})$ , 則  $\phi_p(\mathbf{v}_1, \mathbf{w}) = p^{n-d_1}(1-p)^{d_1}$ ,  $\phi_p(\mathbf{v}_2, \mathbf{w}) = p^{n-d_2}(1-p)^{d_2}$ 。比較兩式即可得證。 ■

利用機率的概念，解碼的概念也就形成了。

(1) 完全最大可能性 (Complete Maximum Likelihood Decoding, CMLD)

如果接收到的字為  $\mathbf{w}$ ，而  $\mathbf{v} \in C$  為唯一最接近  $\mathbf{w}$  的字碼，這時候就選  $\mathbf{v}$  為傳送的字；如果與  $\mathbf{w}$  最接近的碼字不是唯一，則任意選一個當傳送的字。

(2) 不完全最大可能性 (Incomplete Maximum Likelihood Decoding, IMLD)

不同於 (1) 的部分是在有數個（兩個以上）最接近的碼字時，要求重送。

所以，在可以要求重送的情況下，錯誤產生的機會也會小很多；然而，在現實的狀況下，幾乎都是在無法要求重送的前提之下進行，這個時候，產生錯誤時都能找到唯一最接近的碼字就是能加以更正的保證。

再來，我們討論 MLD 的可靠性。

**定義 2.30.**  $\theta_p(C, \mathbf{v})$  = 在  $C$  中，傳送  $\mathbf{v}$  且正確地收到  $\mathbf{v}$  的機率。

所以  $\theta_p(C, \mathbf{v}) = \sum_{\mathbf{w} \in L(\mathbf{v})} \phi_p(\mathbf{v}, \mathbf{w})$ ,

這裡的  $L(\mathbf{v}) = \{\mathbf{z} \in K^n \mid d(\mathbf{z}, \mathbf{v}) < d(\mathbf{z}, \mathbf{y}), \mathbf{y} \in C \setminus \{\mathbf{v}\}\}$ 。

也就是說，在  $L(\mathbf{v})$  中的任一個字都會被更正為  $\mathbf{v}$ ，而且，這個選擇是正確的。

**例 1.**  $C_1 = \{000, 111\}$ ,  $p = 0.9$ ,  $L(000) = \{000, 001, 010, 100\}$

$$\begin{aligned} \theta_p(C_1, 000) &= \phi_P(000, 000) + \phi_P(000, 001) + \phi_P(000, 010) + \phi_P(000, 100) \\ &= p^3 + 3p^2(1-p) = 0.972 \end{aligned}$$

**例 2.**  $C_2 = \{0000, 1010, 0111\}$

$$\theta_p(C_2, 0000) = 0.8019, \theta_p(C_2, 0111) = 0.9558$$

$$(L(0000) = \{0000, 0100, 0001\}, L(0111) = \{0110, 0101, 0011, 1101, 0111, 1111\})$$

由上面的例子，我們可以發現，如果數碼的設計不好，可能會造成有些碼字的傳送正確性較高，而這些碼字傳送的正確性較低；或者整體而言正確性下降，例如將  $C_1$  改成  $C_3 = \{000, 011\}$ ，於是， $L(000) = \{000, 100\}$  而  $L(011) = \{011, 111\}$ 。因此，正確性下降，也表示比較不可靠。

**定義 2.31.** 傳錯率 (Word Error Rate)

$$P_{err}(\mathbf{v}) = 1 - \theta_p(C, \mathbf{v})。$$

(註) 前面所建構的完美碼， $n = 7$ ， $d = 3$ ，傳錯率為  $1 - (1 - p)^7 - 7(1 - p)^6p$  (?)。

由前面的討論，對於如何去解碼已有初步的概念，以下是比較詳細的說明。

**定義 2.32.** 錯誤形式 (Error Pattern)

如果傳送的  $\mathbf{v}$  在收到時是  $\mathbf{w}$ ，則  $\mathbf{e} = \mathbf{v} + \mathbf{w}$  稱為是傳送錯誤形式。

**定義 2.33.**  $t$ -錯偵測碼 (t-error Detecting Code)

能夠偵測出加權小於  $t$  的所有錯誤形式之數碼，稱為是  $t$ -錯偵測碼。

**定義 2.34.**  $t$ -錯更正碼 (t-error Correcting Code)

可以更正加權小於  $t$  的所有錯誤形式之數碼，稱為是  $t$ -錯更正碼。

**定理 2.35.** 如果數碼  $C$  的距離為  $d$ ，則  $C$  為  $(d - 1)$ -錯偵測碼，也是  $\lfloor \frac{d-1}{2} \rfloor$ -錯更正碼。

**證明.** 直觀。 ■

在這一節的最後再介紹線性碼解碼的概念。

前面提到，如果  $C$  是線性碼，則必存在一個核對矩陣  $H$ ，使得對於任意的  $\mathbf{v} \in C$ ， $\mathbf{v}H = \bar{0}$ ，所以，當  $\mathbf{w}$  為收到的字時，首先要做的工作是看  $\mathbf{w}H$  會不會是  $\bar{0}$ ， $\mathbf{w}H$  於是被稱為是徵兆 (Syndrome)。  $\mathbf{w}H$  是一個  $(n - k)$ -維的向量，當然就有  $2^{n-k}$  種可能，這個現象恰好與  $K^n/C$  有  $2^{n-k}$  餘集相呼應。例如，所有在  $C$  中的碼字  $\mathbf{v}$ ， $\mathbf{v}H = \bar{0}$ ，而所有在  $\mathbf{w} + C$  中的字  $\mathbf{z}$ ， $\mathbf{z}H = \mathbf{w}H$ 。

**定理 2.36.** 令  $C$  為一線性碼， $H$  為  $C$  的核對矩陣，如果傳送  $\mathbf{v} \in C$  所造成的錯誤形式為  $\mathbf{e}$ ，則收到的  $\mathbf{w} = \mathbf{v} + \mathbf{e}$ ，而且  $\mathbf{w}H = \mathbf{e}H$ ，亦即  $\mathbf{w}$  與  $\mathbf{e}$  有相同的徵兆。

因此，當  $\mathbf{w}$  為收到的字時，只要在  $\mathbf{w} + C$  這個餘集中找一個加權最小的向量 (Coset Leader) 即為錯誤形式 (可能)。當錯誤形式可以唯一決定時，正確的碼字就可以順利找到。所以，要順利解碼，每一個  $K^n/C$  的餘集都要有唯一的最輕向量 (C.L.)。

這種解碼方式的一個缺點是要儲存所有的餘集相當浪費空間；幸運的是我們在線性碼中不需存這麼多分割好的餘集，在收到  $\mathbf{w}$  之後再去看  $\mathbf{w} + C$  即可；倒是如何保證有唯一的 C.L. 要花一點力氣。

### 3. 一些好的數碼

從前面的討論，我們知道如果  $C$  是長度為  $n$  的數碼，則這個碼能夠傳遞訊息的比例大約是  $\log_2 |C|/n$ ，也就是說在傳遞  $n$  個字母中，有  $\log_2 |C|$  左右的字母代表訊息；於是當  $C$  為線性碼而且  $C$  的維數是  $k$  的時候， $\frac{k}{n}$  就是數碼  $C$  的訊息率。

顯然，訊息率高的數碼比較節省，然而當  $k$  與  $n$  接近時，這個數碼的距離就不容易大，也就是說會產生錯誤的機率  $P_{err} = \sum_{i=1}^n A_i (1-p)^i p^{n-i}$  ( $A_i$  是加權為  $i$  的碼字數) 也會比較大。要使得  $\frac{k}{n}$  夠大，而  $P_{err}$  可以任意小，是研究編碼理論的基本精神，而這件事成立才使得尋找好的數碼 (Good Codes) 有一線曙光。

**定理 3.1.** 令  $p$  為傳遞正確的機率， $q = 1 - p$  為傳錯字母的機率。則對於任意的  $\varepsilon > 0$ ，如果  $R < C(q) = 1 + q \log_2 q + p \log_2 p$  以及  $n$  足夠大就一定會存在一個二元線性碼，它的訊息率  $\frac{k}{n} \geq R$  而且  $P_{err} < \varepsilon$ 。

**證明.** 略。

上述的  $C(q)$  稱為是 BSC 的容量 (Capacity)。

值得一提的是，Shannon 証明好的數碼的確存在，而且是線性碼；可是，他用的是機率式證明，而不是真正去建構一個數碼，滿足所給定的條件，這也才留下一片天空給後世的編碼理論專家去研究，現在離開定理的證明已經超過半世紀，我們究竟做了多少事呢？在這一節中，我們先看兩類出名的數碼。

**定義 3.2.** 漢明碼 (Hamming Codes)

令  $H$  的列由所有不為  $\bar{0}$  的  $r$ -維向量所組成，即  $H$  為  $(2^r - 1) \times r$  的矩陣。則以  $H$  為核對矩陣的數碼稱為是長度  $2^r - 1$  的漢明碼，以  $H_r$  表示。

**命題 3.3.**  $H_r$  為  $(2^r - 1, 2^r - 1 - r, 3)$ -碼。

**證明.**  $n$  與  $k$  很容易看，而  $d$  則由列與列之間的相依關係可以看出。 ■

**命題 3.4.** 對於所有的  $r \geq 2$ ,  $H_r$  為完美碼。

**證明.**

$$\frac{2^{2^r-1}}{\binom{2^r-1}{0} + \binom{2^r-1}{1}} = 2^{2^r-1-r}。$$

■

上節所舉的例子為  $r = 3$  的情況。當  $H = \begin{bmatrix} 1110100 \\ 0111010 \\ 0011101 \end{bmatrix}^t$  時,  $C$  就是由

組合設計所建構出來的那一個。

**定義 3.5.** 等價碼 (Equivalent Codes)

如果兩個碼在經過碼字的字母序交換之後, 則這兩個數碼等價。

**例.**

$$C_1 : \begin{matrix} 0000 \\ 0011 \\ 1100 \\ 1111 \end{matrix} \quad \text{and} \quad C_2 : \begin{matrix} 0000 \\ 0101 \\ 1010 \\ 1111 \end{matrix}。$$

**定理 3.6.** 所有的線性  $(2^r - 1, 2^r - 1 - r, 3)$ -碼皆與  $H_r$  等價。

**證明.** 經過列的重排之後, 可以讓它們具有相同的核對矩陣。

■

**定理 3.7.** 解碼 (Decoding)

假設收到的字是  $\mathbf{y}$ , 而且知道最多只有一個字碼錯, 則在  $\mathbf{y}H \neq \bar{0}$  時,  $H$  中必存在有一列  $H_l$  (第  $l$  列) 使得  $\mathbf{y}H = H_l$ , 此時可以推得第  $l$  個位置是錯誤的地方。

**證明.** 因為  $(\mathbf{y} - \mathbf{e}_l)H = \mathbf{y}H + H_l = \bar{0}$ , 所以  $\mathbf{y} - \mathbf{e}_l \in C$ 。

所以  $H_r$  可以更正一個錯誤。同時,  $H_r$  可以偵測出兩個錯誤。

■

**定義 3.8.** 擴展碼 (Extended Code)

令  $C$  為一  $(n, k, d)$ -碼, 其中  $d$  為奇數。現在將  $C$  中的碼字各加長一個字母使得每個字的加權為偶數, 則所得到的數碼為  $(n + 1, k, d + 1)$ , 此為  $C$  的擴展碼, 以  $\bar{C}$  表示。

所以  $\overline{H_r}$  為  $(2^r, 2^r - 1 - r, 4)$ -碼，它可以多偵測出一個錯誤。

例.  $\overline{H_3}$  是一個  $(8, 4, 4)$ -碼，它的核對矩陣為  $\hat{H} = \begin{bmatrix} 11111111 \\ 00011110 \\ 01100110 \\ 10101010 \end{bmatrix}^t$ 。

**定理 3.9.**  $\overline{H_3}$  為一自對偶碼 (Self-Dual)。

**證明.**  $\hat{H}$  中的四個行向量都是  $\overline{H_3}$  的碼字。 ■

接下來討論如何由已知的數碼經由簡單的變化而獲得另一個新的數碼。

(1) 加入一個核對字母使得每個字的加權都是偶數。

這個動作在原數碼的距離是奇數時比較常見。

(2) 去掉一個字母以減短原來碼字的長度。

去掉的位置是固定的。

(3) 去掉一些碼字 (Expurgating Codewords)。

(4) 加入一些碼字 (Augmenting Codewords)。

(5) 加入  $\bar{1} = 1111 \cdots 1$  以及加入核對字母。

(6) 去掉某一些碼字具有相同字母的那一個位置。

有了上述的改變方式，一個漢明碼就可以被修改成多個不同而且漂亮的數碼，如下圖所示。在這之前，先介紹一種特殊碼。

**定義 3.10.** 等距碼 (Simplex Codes)

一個漢明碼  $H_r$  的對偶碼  $S_r$  稱為是等距碼。

(註) 這個數碼的特性是任兩個碼字的距離都是一個固定數。

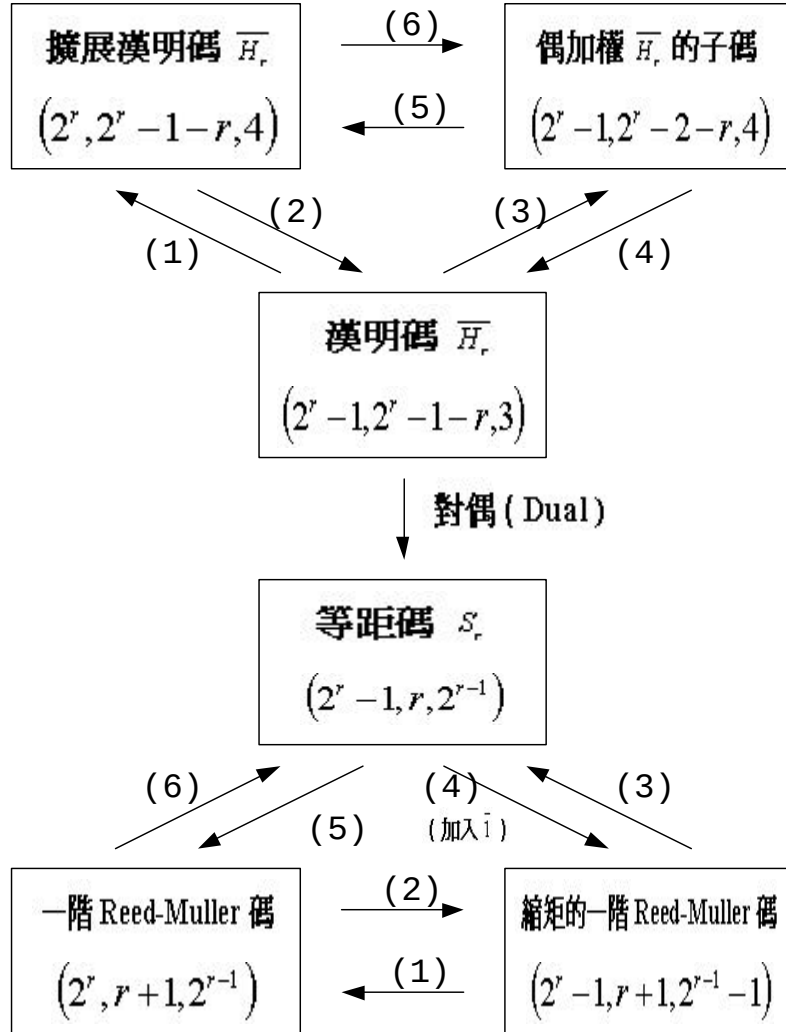
例.

$S_2 = \{000, 011, 101, 110\}$ ， $d(S_2) = 2$ ，而且任意兩個碼字的距離也是 2。

$S_3 = \{0000000, 0110011, 1010101, 1100110, 0001111, 0111100, 1011010, 1101001\}$ ， $d(S_3) = 4$ ，而且任意兩個碼字的距離也是 4。

如果令  $S_r = \{ \mathbf{v} \mid \mathbf{v} + \bar{1} \in H_r \}$  , 則上面所提到的性質不難利用歸納法加以證明。

$$S_{r+1} : \begin{array}{c|c|c} & 0 & S_r \\ & \vdots & \\ & 0 & \\ \hline S_r & 1 & \\ & \vdots & S_r + \bar{1} \\ & 1 & \end{array} \qquad S_3 : \begin{array}{c|c|c} 000 & 0 & 000 \\ 011 & 0 & 011 \\ 101 & 0 & 101 \\ 110 & 0 & 110 \\ \hline 000 & 1 & 111 \\ 011 & 1 & 100 \\ 101 & 1 & 010 \\ 110 & 1 & 001 \end{array}$$



除了上述由一個數碼經過幾種不同的處理方法而得到不同的數碼，我們也可以利用直接把兩個數碼放在一起的概念而製造出新的數碼。以下我們用  $|\mathbf{u}|\mathbf{v}|$  來表示將  $\mathbf{u} \in C_1$  及  $\mathbf{v} \in C_2$  兩個碼字接在一起，新的碼字長度為  $C_1$  的長度加  $C_2$  的長度。

**定義 3.11.** 數碼的和 (Direct Sum)

令  $C_1$  與  $C_2$  分別為長度  $m$  與  $n$  的兩個數碼，則  $C = \{|\mathbf{u}|\mathbf{v}| \mid \mathbf{u} \in C_1, \mathbf{v} \in C_2\}$  為  $C_1$  與  $C_2$  的和，以  $C = C_1 \oplus C_2$  表示。

**命題 3.12.**  $C_1 \oplus C_2$  為  $(m+n, |C_1| \cdot |C_2|, \min\{d(C_1), d(C_2)\})$ -碼。

(註) 因為  $C_1 \oplus C_2$  不一定是線性碼，所以用  $C$  的碼字數代替線性碼中的  $k$ 。這個表示法有時會造成困擾，因此是否為線性碼要特別標示出來。

另一個類似的建構法直接以  $|\mathbf{u}|\mathbf{u} + \mathbf{v}|$  表示，此時， $\mathbf{u}$  與  $\mathbf{v}$  的長度要相同才可以。

**命題 3.13.** 令  $C_1$  與  $C_2$  分別為非線性的  $(n, M_1, d_1)$ -碼及  $(n, M_2, d_2)$ -碼，則  $C_3 = \{|\mathbf{u}|\mathbf{u} + \mathbf{v}| \mid \mathbf{u} \in C_1, \mathbf{v} \in C_2\}$  為非線性的  $(2n, M_1 M_2, \min\{2d_1, d_2\})$ -碼。

**證明.** 除了距離之外，其它的性質都很明顯。現在我們來計算  $d(C_3)$ 。

令  $\mathbf{a} = |\mathbf{u}|\mathbf{u} + \mathbf{v}|$ ， $\mathbf{b} = |\mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$  為  $C_3$  中相異的兩個碼字。如果  $\mathbf{v} = \mathbf{v}'$ ，則  $d(\mathbf{a}, \mathbf{b}) = 2d(\mathbf{u}, \mathbf{u}') \geq 2d_1$ 。在另一方面，如果  $\mathbf{v} \neq \mathbf{v}'$ ，則  $d(\mathbf{a}, \mathbf{b}) = wt(\mathbf{u} + \mathbf{u}') + wt(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}') \geq wt(\mathbf{v} + \mathbf{v}') \geq d_2$ 。 ■

利用上述的概念我們將三個數碼接在一起。令  $H_3^* = \{\mathbf{v} = x_1 x_2 x_3 \cdots x_7 \mid x_7 x_6 x_5 \cdots x_1 \in H_3\}$ ，也就是把碼字位置次序反轉過來所得到的數碼。由於  $H_3^*$  與  $H_3$  等價，所以它們都是  $(7, 4, 3)$ -線性碼，於是它們的擴展碼  $\overline{H_3^*}$  與  $\overline{H_3}$  分別為  $(8, 4, 4)$ -線性碼。現在把它們接在一起就可以得到一個非常漂亮的數碼。

**定義 3.14.** 擴展的哥雷碼 (Extended Binary Golay Code  $G_{24}$ )

$$G_{24} = \{|\mathbf{a} + \mathbf{x}|\mathbf{b} + \mathbf{x}|\mathbf{a} + \mathbf{b} + \mathbf{x}| \mid \mathbf{a}, \mathbf{b} \in \overline{H_3}, \mathbf{x} \in \overline{H_3^*}\}。$$

**命題 3.15.**  $G_{24}^\perp = G_{24}$  。

**證明.** 令  $\mathbf{a}, \mathbf{b}$  分別為  $\overline{H}_3$  中一組基底的向量,  $\mathbf{x}$  則為  $\overline{H}_3^*$  的一組基底的向量, 則不難驗證  $\{|\mathbf{a}|\overline{0}|\mathbf{a}|, |\overline{0}|\mathbf{b}|\mathbf{b}|, |\mathbf{x}|\mathbf{x}|\mathbf{x}|\}$  中共 12 個  $G_{24}$  的碼字, 它形成  $G_{24}$  的一組基底; 因為每個碼字的加權均為偶數, 而且此為垂直基底 (Orthogonal Basis), 因此  $G_{24}^\perp = G_{24}$  (?) 。

**命題 3.16.**  $d(G_{24}) = 8$  。

**證明.** 因為基底中的碼字它們的加權均為 4 的倍數, 所以  $4|d(G_{24})$ 。假設  $d(G_{24}) = 4$ , 則必存在一個不是  $\overline{0}$  的碼字,  $\mathbf{w} = |\mathbf{a} + \mathbf{x}|\mathbf{b} + \mathbf{x}|\mathbf{a} + \mathbf{b} + \mathbf{x}|$ , 它的加權為 4。由於  $\mathbf{w}$  的每一小部分 (三部分) 的加權均為偶數, 所以, 必定有一部分是  $\overline{0}$ , 這表示  $\mathbf{x} = \overline{0}$  or  $\overline{1}$  (?)。假設  $\mathbf{x} = \overline{0}$  ( $\overline{1}$  的討論相似)。於是,  $|\mathbf{a}|\mathbf{b}|\mathbf{a} + \mathbf{b}|$  的加權為 4, 這是不可能的, 所以得證。

**命題 3.17.** 令  $G_{23}$  為  $G_{24}$  的縮短碼 (去掉一個字母), 則  $G_{23}$  為一完美碼。

**證明.**

$$\frac{2^{23}}{\binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3}} = 2^{12} \text{。} (d(G_{23}) = 7)$$

(註) 把  $G_{24}$  中加權為 8 的碼字轉換成集合, 這個集合族有 759 集合, 它們形成一個  $5 - (24, 8, 1)$  設計 (5-design) (?)。

除了上述建構  $G_{24}$  的方法, 另外可以用二次同餘 (Quadratic Residue) 的方式來建構。令  $B'$  為  $11 \times 11$  的方陣, 如下:

$$B' = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(註) 第一列不為 0 的位置是 1, 2, 4, 5, 6, 10, 這 6 個數在  $Z_{11}$  中是  $x^2$  可能和它同餘的數。

$$\text{再令 } B = \begin{bmatrix} & & & & & & & & & 1 \\ & & & & & & & & & 1 \\ & & B' & & & & & & & 1 \\ & & & & & & & & & \vdots \\ & & & & & & & & & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}, \text{ 及 } G = [B|I_{12}].$$

於是，利用  $G$  為生成矩陣，我們可以建構出一個  $(24, 12, 8)$ -碼  $G_{24}$ ，這裡的距離是 8 需要一些努力，讀者請試試看。值得一提的是，再由去掉最後一個位置同樣可以獲得  $(23, 12, 7)$ -碼；但是，它與前面建構的  $G_{23}$  等價嗎？它的答案是肯定，請參考 McWilliams 及 Sloane 那本書；基本上，S.L.Snover 在 1973 年證明任意的  $(23, 2^{12}, 7)$ -碼（不一定是線性碼）都會與  $G_{23}$  等價。

對於擴展的哥雷碼  $G_{24}$  的瞭解，可以由利用加權算子來探討。

**定義 3.18.** 加權算子 (Weight Enumerator)

令  $C$  為一長度為  $n$  的線性碼， $A_i$  為在  $C$  中加權為  $i$  的碼字數，則  $A(z) = \sum_{i=0}^n A_i z^i$  被定義為  $C$  的加權算子，而  $\{A_i\}_{i=0}^n$  則稱為是  $C$  的加權分佈 (Weight Distribution)。

由於  $d(C) = 8$ ，所以  $A_i = 0$ ， $i = 0, 1, 2, \dots, 7$ 。至於  $A_8$  以上就得花一翻功夫。為了方便計算，我們選擇  $G_{24}$  的一個生成矩陣為

$$G = \begin{bmatrix} 1 & & & & & & & & & 0 \\ 1 & & & & & & & & & 0 \\ 1 & & I_{11} & & & & & & 0 & B' \\ \vdots & & & & & & & & \vdots & \\ 1 & & & & & & & & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

所以每一個  $G_{24}$  的碼字可以表示成  $|L| R|$ 。

**命題 3.19.** 假如  $|L| |R| \in G_{24}$ ,  $L = l_1 l_2 \cdots l_{12}$ , 以及  $R = r_1 r_2 \cdots r_{12}$ , 則  $|L'| |R'| \in G_{24}$ , 其中  $L' = r_1 r_2 r_{12} r_{11} \cdots r_3$ ,  $R' = l_1 l_2 l_{12} l_{11} \cdots l_3$ 。

**證明.** 令  $\varphi = (l_1 r_1)(l_2 r_2)(l_{12} r_3)(l_{11} r_4) \cdots (l_3 r_{12})$ , 則  $G$  的第一列  $|110000000000|011011100010|$  對應成  $|0101000111011|110000000000|$ 。這個字恰好是以下 7 列 1, 3, 7, 8, 9, 11, 12 的和; 同理可推得第 2 列至第 11 列皆可以對應成其它數列之和, 所以是碼字, 而最後一列則對應為  $|\bar{1} | \bar{0} |$ , 它是最後一列加  $\bar{1}$ , 由於  $\bar{1} \in G_{24}$ , 所以它也是碼字。 ■

由於  $G_{24}$  中的碼字它的加權均為 4 的倍數, 所以只有可能在  $i = 0, 4, 8, 12, 16, 20, 24$  時  $A_i \neq 0_i$ ; 由命題 3.19, 如果  $|L| |R| \in G_{24}$ ,  $wt(L) = i$ ,  $wt(R) = j$ , 則  $G_{24}$  中必定有一個碼字  $|L'| |R'|$ , 它滿足  $wt(L') = j$ ,  $wt(R') = i$ 。於是, 我們不難推得下面的結論。

**命題 3.20.**  $G_{24}$  中沒有加權為 4 與 20 的碼字。

**證明.**(略)。

所以各種加權的碼字可以順利的計算出來: 概念是先由前面 11 列來找出碼字, 若有需要再加上最後一列, 所以  $|L| |R| + |\bar{0} | \bar{1}| = |L| |\bar{R}|$ 。

$$\begin{aligned} A_{16} = A_8 &= 2((2, 6)\text{-型}) + 2((6, 2)\text{-型}) + \binom{11}{3} + \binom{11}{4} \\ &= 4 \left[ \binom{11}{1} + \binom{11}{2} \right] + \binom{11}{3} + \binom{11}{4} = 759。 \end{aligned}$$

$$A_{12} = 2^{12} - 2 - 2 \times 759 = 2576, \text{ 於是對於 } C = G_{24},$$

$$A(z) = 1 + 759z^8 + 2576z^{12} + 759z^{16} + z^{24}。$$

現在再來看當  $C = H_r$  時的加權算子。為了方便起見, 令  $2^r - 1 = n$ 。由漢明碼的特性, 如果我們選  $H$  中的  $i - 1$  列來求和, 則令有以下三種可能: (1) 和為另一列, (2) 和為  $i - 1$  列中的一列, 以及 (3) 和為  $\bar{0}$ , 所以依這三種可能求得

$$\binom{n}{i-1} = iA_i + (n-i+2)A_{i-2} + A_{i-1}。$$

兩邊同乘  $z^{i-1}$ , 並且由  $i = 0, 1, \dots$  把全部加起來。

$$\sum_{i=0}^{\infty} \binom{n}{i-1} z^{i-1} = \sum_{i=0}^{\infty} [iA_i z^{i-1} + (n-i+2)A_{i-2} z^{i-1} + A_{i-1} z^{i-1}]$$

$$\begin{aligned} (1+z)^n &= (A_1 + 2A_2 z + 3A_3 z^2 + \cdots + nA_n z^{n-1}) \\ &\quad + n(A_0 z + A_1 z^2 + A_2 z^3 + \cdots + A_{n-1} z^n + A_n z^{n+1}) \\ &\quad - (A_1 z^2 + 2A_2 z^3 + 3A_3 z^4 + \cdots + A_{n-1} z^n + A_n z^{n+1}) \\ &\quad + (A_0 + A_1 z + \cdots + A_n z^n) \end{aligned}$$

$$(1+z)^n = A'(z) + nzA(z) - z^2 A'(z) + A(z),$$

$$\text{所以 } A(z) \cdot (1+nz) = -(1-z^2)A'(z) + (1+z)^n,$$

解微分方程,  $A_0 = 1$ 。

$$\text{因此, } A(z) = \frac{1}{n+1} (1+z)^n + \frac{n}{n+1} (1+z)^{\frac{n-1}{2}} (1-z)^{\frac{n+1}{2}} \text{。 (?)}$$

以  $n = 7$  為例,

$$\begin{aligned} A(z) &= \frac{1}{8}(1+z)^7 + \frac{7}{8}(1+z)^3(1-z)^4 \\ &= \frac{1}{8}(1+7z+21z^2+35z^3+35z^4+21z^5+7z^6+z^7) \\ &\quad + \frac{7}{8}(1-3z^2+3z^4-z^6-z+3z^3-3z^5+z^7) \\ &= \frac{1}{8}(8+56z^3+56z^4+8z^7) \\ &= 1+7z^3+7z^4+z^7 \end{aligned}$$

這個式子告訴我們在  $H_3$  中, 只有加權為 0, 3, 4, 7 的碼字, 而它們分別是 1, 7, 7, 1 個。

而這一節最後我們介紹哥雷碼如何解碼。

令  $H = \begin{bmatrix} I_{12} \\ B \end{bmatrix}$ , 送  $\mathbf{v}$ , 收到  $\mathbf{w}$ , 錯誤型為  $\mathbf{e}$ , 並且令  $\mathbf{e}_i$  為長度 12 的字, 除了第  $i$  個位置為 1, 其它位置都是 0; 而  $\mathbf{b}_i$  則是  $B$  的第  $i$  列。

解碼的演算法 (Decoding Algorithm, IMLD)

1. 計算  $\mathbf{s} = \mathbf{w}H$ 。
2. 假如  $wt(\mathbf{s}) \leq 3$ , 則  $\mathbf{e} = |\mathbf{s}| \bar{0}|$ 。
3. 假如對於某個  $i, i \in \{1, 2, \dots, 12\}$ ,  $wt(\mathbf{s} + \mathbf{b}_i) \leq 2$ , 則  $\mathbf{e} = |\mathbf{s} + \mathbf{b}_i| \mathbf{e}_i|$ 。
4. 計算  $\mathbf{s}B$ 。

5. 假如  $wt(\mathbf{s}B) \leq 3$  , 則  $\mathbf{e} = |\bar{0}| \mathbf{s}B|$  。
6. 假如對於某個  $i$  ,  $i \in \{1, 2, \dots, 12\}$  ,  $wt(\mathbf{s}B + \mathbf{b}_i) \leq 2$  , 則  $\mathbf{e} = |\mathbf{e}_i| \mathbf{s}B + \mathbf{b}_i|$  。
7. 如果無法決定 , 則要求重送 。

令  $\mathbf{e} = |\mathbf{e}'| \mathbf{e}''|$  , 利用核對矩陣  $H = \begin{bmatrix} I \\ B \end{bmatrix}$  , 求得  $\mathbf{s} = \mathbf{w}H = \mathbf{e}H = \mathbf{e}' + \mathbf{e}''B$  。

如果  $wt(\mathbf{e}) \leq 3$  , 則  $\mathbf{e}'$  與  $\mathbf{e}''$  中必有一個它的加權不大於 1 。

- (i)  $wt(\mathbf{e}'') = 0$  , 則  $\mathbf{s} = \mathbf{e}'$  , 因此  $|\mathbf{s}| \bar{0}|$  為錯誤型 , 這種情況在  $wt(\mathbf{s}) \leq 3$  時產生 。
- (ii)  $wt(\mathbf{e}'') = 1$  , 則  $\mathbf{s} = \mathbf{e}' + \mathbf{b}_i$  , 因此對於某個  $i$  ,  $i \in \{1, 2, \dots, 12\}$  ,  $wt(\mathbf{s} + \mathbf{b}_i) \leq 2$  , 此時錯誤型為  $|\mathbf{s} + \mathbf{b}_i| \mathbf{e}''|$  。
- (iii) 如果不是 (i) 與 (ii) , 則  $wt(\mathbf{e}') \leq 1$  , 此時要利用  $\mathbf{s}B$  。

$\mathbf{s}B = (\mathbf{e}' + \mathbf{e}''B)B = \mathbf{e}'B + \mathbf{e}''(B^2 = I)$  , 這個形式與前面的  $\mathbf{s}$  對應 。

- (iv) 如果  $wt(\mathbf{e}') = 0$  , 則錯誤型  $\mathbf{e} = |\bar{0}| \mathbf{s}B|$  。(  $wt(\mathbf{s}B) \leq 3$  )
- (v) 如果  $wt(\mathbf{e}') = 1$  , 則存在一個  $i$  , 使得  $wt(\mathbf{s}B + \mathbf{b}_i) \leq 2$  , 錯誤型為  $|\mathbf{e}_i| \mathbf{s}B + \mathbf{b}_i|$  。
- (vi) 若是無法決定 , 就只好要求重送 。

例. 傳  $\mathbf{v}$  收到  $\mathbf{w} = |001001001101|101000101000|$  。

$$\mathbf{s} = 110001001001$$

$wt(\mathbf{s}) = 5$  , 跳過第 2 步 , 找一個  $\mathbf{b}_i$  使得  $wt(\mathbf{s} + \mathbf{b}_i) \leq 2$  ,  $i = 5$  ,  
 $\mathbf{s} + \mathbf{b}_i = 000000010010$  , 所以錯誤為  $|000000010010|000010000000|$

$$\mathbf{v} = |001001011111|101010101000|$$

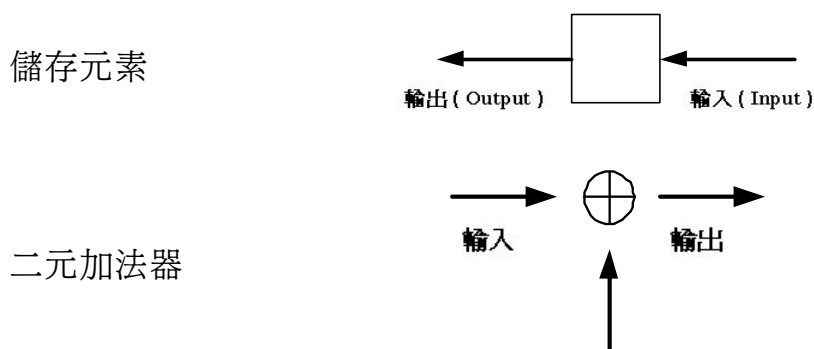
**Problem.** Decode  $|000111000111|011011010000|$  。

解:  $\mathbf{v} = |000011000111|011010000000|$  。

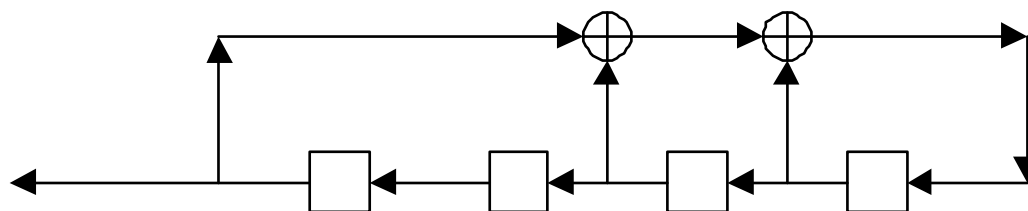
## 4. 循環碼 (Cyclic Codes)

由於在邊碼的過程中可以使用所謂的平移登陸 (Shift Register)，所以循環碼成為最熱門的數碼；幾乎所有好的數碼都是循環碼，包括前面介紹的漢明碼及哥雷碼。

首先介紹平移登陸器；它由兩部分組合而成：



例.



|                  |       |       |                   |                   |
|------------------|-------|-------|-------------------|-------------------|
| $t_0$ : Time "0" | $a_0$ | $a_1$ | $a_2$             | $a_3$             |
| $t_1$ : Time "1" | $a_1$ | $a_2$ | $a_3$             | $a_0 + a_2 + a_3$ |
| $t_2$ : Time "2" | $a_2$ | $a_3$ | $a_0 + a_2 + a_3$ | $a_1 + a_0 + a_2$ |

所以，如果一開始  $(a_0, a_1, a_2, a_3) = (1, 1, 0, 1)$

|            |   |   |   |   |   |   |          |
|------------|---|---|---|---|---|---|----------|
| $t_0$ :    |   |   | 1 | 1 | 0 | 1 |          |
| $t_1$ :    |   |   | 1 | 1 | 0 | 1 | 0        |
| $t_2$ :    |   |   | 1 | 1 | 0 | 1 | 0        |
| $t_3$ :    | 1 | 1 | 0 | 1 | 0 | 0 | 0        |
| $t_4$ :    | 1 | 1 | 0 | 1 | 0 | 0 | 0        |
|            |   |   |   |   |   |   | 1        |
|            |   |   |   |   |   |   | $\vdots$ |
| $t_{10}$ : | 1 | 1 | 0 | 1 | 0 | 0 | 0        |

如此一來，就可以不斷地生成連續 7 個數字的碼字，循環碼的概念就形成了。

**定義 4.1.** 循環平移 (Cyclic Shift)

令  $\mathbf{v} = (a_0, a_1, a_2, \dots, a_{n-1})$ ，則  $\mathbf{v}$  的循環平移  $\pi(\mathbf{v}) = (a_{n-1}, a_0, a_1, \dots, a_{n-2})$ 。因此  $\pi^n(\mathbf{v}) = \mathbf{v}$ 。

**定義 4.2.** 循環碼 (Cyclic Codes)

如果在數碼  $C$  中，當  $\mathbf{v} \in C$  則  $\pi(\mathbf{v}) \in C$ ，則  $C$  稱為是循環碼。當  $C$  為線性碼時， $C$  也稱為是線性循環碼 (Linear Cyclic Code)。

研究循環碼最好的方法是把一個向量  $\mathbf{v} = (a_0, a_1, a_2, \dots, a_{n-1})$  用多項式來表示，亦即  $a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1}$ 。由於  $\pi(\mathbf{v}) = x(a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1}) \bmod (x^n - 1) = a_{n-1} + a_0x + a_1x^2 + \dots + a_{n-2}x^{n-1}$ ，再用向量來看就是  $(a_{n-1}, a_0, a_1, \dots, a_{n-2})$ ，它也是  $\pi(\mathbf{v})$ 。因此，循環平移可以看成是多項式的乘法，換言之  $\pi^k(\mathbf{v})$  對應於  $x^k(a_0 + a_1x + \dots + a_{n-1}x^{n-1}) \bmod (x^n - 1)$ 。

以下  $x^n - 1$  可以換成  $x^n + 1$ ，這是因為數  $a_i \in GF(2)$  的關係。

**定理 4.3.** 令  $R_n = F[x]/\langle x^n + 1 \rangle$ ，則  $C \subseteq R_n$  為一線性循環碼的充要條件為  $C$  是  $R_n$  的一個理想。

**證明.**

( $\Leftarrow$ ) 因為  $C$  是一個理想，因此對於當  $f(x) \in C$  時  $x \cdot f(x) \in C$ ，因此  $C$  為循環碼。

( $\Rightarrow$ ) 令  $f(x) \in C$ ， $g(x) \in R_n$  以及  $g(x) = b_0 + b_1x + b_2x^2 + \dots + b_{n-1}x^{n-1}$ ，則  $f(x)g(x) = b_0f(x) + b_1xf(x) + \dots + b_{n-1}x^{n-1}f(x)$  是一些  $f(x)$  的循環平移所組合而成，因此  $f(x)g(x) \in C$ 。加上  $C$  為線性碼， $C$  是  $R_n$  的一個理想。 ■

**定理 4.4.** 令  $C$  為一長度為  $n$  的線性循環碼， $g(x) \neq 0$  為  $C$  中次數最低的多項式，則  $C = \langle g(x) \rangle$ ，同時在  $C$  中沒有其它與  $g(x)$  相同次數的多項式。

**證明.** 利用長除法即可證明任意在  $C$  中的多項式皆為  $g(x)$  的倍式。 ■

**定理 4.5.** 假如  $C$  是線性循環碼而且  $C = \langle g(x) \rangle$ , 則  $g(x) \mid x^n + 1$ 。

**證明.** 利用長除法。 ■

**定理 4.6.** 假如  $g(x)$  的次數為  $n - k$ , 則  $C = \langle g(x) \rangle$  為  $(n, k)$ -碼。同時, 如果  $g(x) = g_0 + g_1x + g_2x^2 + \cdots + g_{n-k}x^{n-k}$ , 則以下的  $G$  為  $C$  的一個生成矩陣。

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & g_0 & g_1 & g_2 & \cdots & g_{n-k} \end{bmatrix}$$

**證明.**  $\{g(x), xg(x), \cdots, x^{k-1}g(x)\}$  為  $C$  的一組基底。

由定理 4.5, 我們要建構循環碼只要將  $x^n + 1$  分解因式即可; 任何一個因式都可以用來生成一個循環碼。 ■

**例.**  $x^7 + 1 = (1 + x)(1 + x + x^3)(1 + x^2 + x^3)$

如果用  $1 + x + x^3$  來生成數碼, 則生成矩陣為

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

然而, 要分解  $x^n + 1$  並不是一件容易的事, 以下介紹另一形式的生成多項式。

**定義 4.7.** 令  $I$  為  $R_n$  中的一個理想,  $C$  為對應於  $I$  的循環碼。我們稱  $C$  的一個多項式  $e(x)$  為等幂生成式 (Idempotent generator) 若且唯若  $c = e(x)$  且  $[e(x)]^2 = e^2(x) = e(x)$ 。

**命題 4.8.** 令  $C = \langle e(x) \rangle$  且  $e(x)$  為等幂生成式, 則  $e(x)$  為  $C$  的乘法單位元素。

**證明.** 令  $a(x) \in C$ , 則  $a(x) = b(x)e(x)$ ,  $b(x) \in R_n$ 。由於  $e^2(x) = e(x)$ , 所以  $a(x)e(x) = b(x)e(x) = a(x)$ 。 ■

究竟要如何才能找到所謂的等冪生成式呢？請看以下的例子。當  $n = 7$  時，令  $e(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_6 x^6$ ，由於  $e^2(x) = e(x)$ ，所以  $e(x) = \alpha_0^2 + \alpha_1^2 x^2 + \alpha_2^2 x^4 + \alpha_4^2 x + \alpha_5^2 x^3 + \alpha_6^2 x^5 + \alpha_3^2 x^6$ 。於是，我們發現，當  $e(x)$  中的  $x^i$  具有非零的係數時， $x^{2i}$  的係數也不是零，這裡是  $2i \pmod{7}$ 。令  $S$  為所有非零係數的次方數，則  $i \in S \Leftrightarrow 2i \pmod{7} \in S$ 。這表示，當  $e(x)$  的一次項係數為 1 時，則  $e(x)$  的二次項，四次項的係數均為 1；同理，在  $e(x)$  中當  $x^3$  的係數為 1 時， $x^6$ 、 $x^5$  的係數也是 1。這個現象使得我們在找  $x^n + 1$  的因子時有很好的參考資訊。

**定義 4.9.** 循環餘集 (Cyclotomic Coset)

$S_1, S_2, \dots, S_k$  稱為是  $Z_n$  的循環餘集若且唯若

- (i)  $\bigcup_{i=1}^k S_i = Z_n$  及
- (ii)  $\forall i \in \{1, 2, \dots, k\}, j \in S_i \Leftrightarrow 2j \pmod{n} \in S_i$ 。

**例.**  $\{0\}, \{1, 2, 4\}, \{3, 5, 6\}$  為  $Z_7$  的循環餘集。

**例.**  $\{0\}, \{1, 2, 4, 8\}, \{3, 6, 9, 12\}, \{5, 10\}, \{7, 14, 13, 11\}$  為  $Z_{15}$  的循環餘集。

**(註)** 循環餘集的個數代表分解  $x^n + 1$  所產生的因子數，而餘集的大小代表該因子的次數。

**例.**  $x^{15} + 1 = (x + 1)(x^2 + x + 1)(x^4 + x + 1)(x^4 + x^3 + 1)(x^4 + x^3 + x^2 + x + 1)$ 。

**定理 4.10.** (輔助定理) 令  $f(x) \in R_n$ ，則  $f(x)$  是等冪多項式的充要條件為  $f(x)$  中，非零係數的次數集可以寫成循環餘集的聯集。

**證明.** 簡單 (略)。

所以要寫出  $n = 7$  時的等冪生成式很簡單，它們是  $0, 1, x + x^2 + x^4, x^3 + x^5 + x^6, 1 + x + x^2 + x^4, 1 + x^3 + x^5 + x^6, x + x^2 + x^3 + x^4 + x^5 + x^6$ ，以及  $1 + x + x^2 + \cdots + x^6$ 。

事實上，任一循環碼除了可以找到生成多項式 (generator) 是天經地義外，找到等冪生成式也沒有任何困難。

**定理 4.11.** 任意長度為奇數的線性循環碼都可以找到一個等冪生成多項式。

**證明.** 令  $C = \langle g(x) \rangle$  為一循環碼。由定理 4.5,  $g(x) | x^n + 1$ , 所以  $x^n + 1 = g(x)h(x)$ , 因為  $n$  是奇數, 所以  $x^n + 1$  的質因式都不相同 (?)。因此  $(g(x), h(x)) = 1$ 。由輾轉相除法, 一定存在兩個多項式  $a(x)$  及  $b(x)$ , 使得

$$a(x)g(x) + b(x)h(x) = 1 \quad (1)$$

令  $e(x) = a(x)g(x)$ , 因此  $e(x) \in C$ 。

由 (1)  $e(x) = a^2(x)g^2(x) + a(x)b(x)g(x)h(x) = a^2(x)g^2(x) = e^2(x) \in R_n$ 。

另外, 設  $c(x) \in C$  為任意的碼字, 則  $c(x) = d(x)g(x)$ , 再利用 (1),  $c(x) = d(x)g(x)a(x)g(x) + d(x)g(x)b(x)h(x) = [d(x)g(x)][a(x)g(x)] = c(x)e(x)$ , 所以  $\langle e(x) \rangle = C$ , 而且  $e(x)$  為  $C$  的單位元素。 ■

**(註)** 由於  $\langle e(x) \rangle = C$ , 我們也可以利用  $e(x)$  來建構生成矩陣, 方法如定理 4.6 所示。

從上面的討論, 我們不難看出一個循環碼  $C$ , 它可以有不同的生成多項式, 一個生成多項式, 一個等冪生成多項式; 以 7 為例子, 它們對應的情形如下:

| 生成多項式                                   | 等冪生成多項式                                            |
|-----------------------------------------|----------------------------------------------------|
| $g_1(x) = (1 + x + x^3)(1 + x^2 + x^3)$ | $e_1(x) = g_1(x)$                                  |
| $g_2(x) = (1 + x)(1 + x + x^3)$         | $e_2(x) = x^3 g_2(x)$                              |
| $g_3(x) = (1 + x)(1 + x^2 + x^3)$       | $e_3(x) = g_3(x)$                                  |
| $g_4(x) = (1 + x)$                      | $e_4(x) = (x + x^3 + x^5)g_4(x) = e_2(x) + e_3(x)$ |
| $g_5(x) = (1 + x + x^3)$                | $e_5(x) = xg_5(x) = e_1(x) + e_2(x)$               |
| $g_6(x) = (1 + x^2 + x^3)$              | $e_6(x) = x^3 g_6(x) = e_1(x) + e_3(x)$            |
| $g_7(x) = 1$                            | $e_7(x) = 1$                                       |

**觀察 :**  $x^7 + 1$  分解成  $1 + x$ 、 $1 + x + x^3$ 、 $1 + x^2 + x^3$  的乘積, 所以生成多項式為這些多項式的組合; 而等冪生成多項式則由  $e_1(x)$ 、 $e_2(x)$  及  $e_3(x)$  所組合, 而且  $(\star) e_1(x) + e_2(x) + e_3(x) = 1$ ,  $e_i(x)e_j(x) = 0$ ,  $1 \leq i \neq j \leq 3$ 。

**定義 4.12.** 數碼和 (Code Sum)

$C_1$  與  $C_2$  兩個數碼和定義為  $C_1 + C_2 = \{c_1(x) + c_2(x) | c_i(x) \in C_i, i = 1, 2\}$ 。

**定理 4.13.** 令  $C_1$  與  $C_2$  為循環碼，它們的生成多項式與等幂生成多項式分別為  $g_1(x)$ 、 $g_2(x)$ ； $e_1(x)$ 、 $e_2(x)$ ；則  $C_1 + C_2$  的生成多項式與等幂生成多項式分別為  $\gcd(g_1(x), g_2(x))$  與  $e_1(x) + e_2(x) - e_1(x)e_2(x)$ 。

**證明.** 我們只證明第二部分： $e(x) = e_1(x) + e_2(x) - e_1(x)e_2(x)$  為  $C_1 + C_2$  的等幂生成多項式；先證明  $e(x)$  為單位元素。令  $c(x) \in C_1 + C_2$  即  $c(x) = c_1(x) + c_2(x)$ 。

$$\begin{aligned} c(x)e(x) &= (c_1(x) + c_2(x))(e_1(x) + e_2(x) - e_1(x)e_2(x)) \\ &= c_1(x)e_1(x) + c_1(x)e_2(x) - c_1(x)e_1(x)e_2(x) \\ &\quad + c_2(x)e_1(x) + c_2(x)e_2(x) - c_2(x)e_1(x)e_2(x) \\ &= c_1(x) + c_1(x)e_2(x) - c_1(x)e_2(x) + c_2(x)e_1(x) + c_2(x) - c_2(x)e_1(x) \\ &= c_1(x) + c_2(x) \end{aligned}$$

所以  $e(x)$  自然會生成  $C_1 + C_2$ ，亦即  $C_1 + C_2 = \langle e(x) \rangle$ 。 ■

有了上述數碼和的概念，我們就可以找到小的循環碼，再由這些小的來建構大的循環碼。

**定義 4.14.** 極小理想 (Minimal Ideal)

一個  $R$  中的理想  $M$ ，如果除了 0 之外沒有任何其它理想會是它的真子理想 (Proper Sub-ideal)，則  $M$  為  $R$  的一個極小理想。

**定理 4.15.** 如果  $f(x) = (x+1)f_1(x)f_2(x)\cdots f_k(x)$  其中  $f_i(x)$  為不可分解的多項式 (佈於  $\text{GF}(2)$ )，則一個循環碼為極小理想的充要條件為它的生成多項式是  $\frac{f(x)}{(x+1)}$  或  $\frac{f(x)}{f_i(x)}$ ， $i = 1, 2, \dots, k$ 。

**證明.** 自行證明。

為了方便表示， $\frac{f(x)}{f_i(x)}$  表成  $\overline{f_i(x)}$ 。

**定理 4.16.** 令  $x^n + 1 = (x+1)f_1(x)f_2(x)\cdots f_k(x)$ ，其中  $f_i(x)$  均不可再分解。則必存在  $k+1$  極小理想分別代表  $k+1$  個循環碼  $C_1, C_2, \dots, C_{k+1}$ ，同時它們的等幂生成多項式分別為  $e_1(x), e_2(x), \dots, e_{k+1}(x)$  且滿足：

- (i)  $e_i(x)e_j(x) = 0$ ， $1 \leq i \neq j \leq k+1$ 。
- (ii)  $\sum_{i=1}^{k+1} e_i(x) = 1$ 。

不但如此，任意的循環碼  $C$  都可以寫成一些  $C_i$  的和，同時它的等幂生成多項式也可以用  $C_i$  的等幂生成多項式的和表示出來。

**證明.** 我們先證明 (i) 和 (ii)。由於  $C_i \cap C_j = 0$ ，所以  $e_i(x)e_j(x) = 0 (i \neq j)$ ，另外因為  $C_1 + C_2$  的等幂生成多項式為  $e_1(x) + e_2(x) + \cdots + e_{k+1}(x)$ 。然而  $C_1 + C_2 + \cdots + C_{k+1} = \langle 1 \rangle$ ，所以 (ii) 得證。

接下來再看第二部分。由於代表  $C_1 + C_2 + \cdots + C_{k+1}$  的極小理想分別為  $\langle \overline{x+1} \rangle = \langle \overline{f_0(x)} \rangle$ ， $\langle \overline{f_1(x)} \rangle$ ， $\cdots$ ， $\langle \overline{f_k(x)} \rangle$ ，因此任意的循環碼  $C = \langle g(x) \rangle$ ， $C = \sum_{j \in I} C_j$ ，其中  $I \subseteq \{0, 1, \cdots, k\}$  及  $(f_i(x), g(x)) = 1$ 。至於  $e(x)$  則為  $\sum_{j \in I'} e_j(x)$ ，其中  $I' \subseteq \{0, 1, \cdots, k\}$  及  $C_j \subseteq C$ 。 $(C = \sum_{C_j \subseteq C} C_j)$  ■

由前面的描述，我們不難發現以下的性質：

**定理 4.17.** 如果  $e(x)$  為循環碼  $C$  的等幂生成多項式，則  $C$  的生成多項式  $g(x) = g.c.d(e(x), x^n + 1)$ 。

**證明.** 由定理 4.11 的證明我們知道  $a(x)g(x) + b(x)h(x) = 1$  中的  $a(x)g(x) = e(x)$ ，而且  $(a(x), h(x)) = 1$ ，所以  $(e(x), g(x)h(x)) = (a(x)g(x), h(x)g(x)) = g(x)$ 。 ■

由  $C$  的等幂生成多項式  $e(x)$ ，也可以求出  $C^\perp$  的等幂生成多項式。

**定理 4.18.** 若是  $x^n + 1 = g(x)h(x)$ ， $C = \langle g(x) \rangle$ ， $C' = \langle h(x) \rangle$ ， $e(x)$  為  $C$  的等幂生成多項式，則  $C'$  的等幂生成多項式為  $1 - e(x)$ ，而  $C^\perp$  的等幂生成多項式為  $1 - e(x^{-1})$ 。

**證明.**  $\forall c(x) \in \langle h(x) \rangle$ ，

$$\begin{aligned} c(x)(1 - e(x)) &= c(x) - c(x)e(x) \\ &= c(x) - c(x)a(x)g(x) \quad (\text{定理 4.10}) \\ &= c(x) \quad (h(x) | c(x)) \end{aligned}$$

所以  $1 - e(x)$  為單位元素 (unit)，自然是等幂生成多項式。另一結果是因為  $C^\perp = \langle x^k h(x^{-1}) \rangle$ ，所以， $\forall c(x) \in C^\perp$ ， $c(x) = d(x)x^k h(x^{-1})$ ， $c(x)(1 - e(x^{-1})) = c(x) - c(x)e(x^{-1})$   
 $= c(x) - d(x)x^k h(x^{-1})e(x^{-1}) = c(x) - d(x)x^k a(x^{-1})x^{-n}(1 + x^n) = c(x)$ 。 ■

在這章最後，我們來看看循環碼的解碼問題。

**定義 4.19.** 徵候多項式 (Syndrome Polynomial)

令  $C = \langle g(x) \rangle$  , 則收到的多項式 (碼字)  $w(x)$  的徵候多項式定義為  $s(x) \equiv w(x) \pmod{g(x)}$  , 亦即用  $w(x)$  除以  $g(x)$  所得到的餘式。

由於  $\deg(g(x)) = n - k$  , 所以  $s(x)$  的次數必小於  $n - k$  , 也就是說  $s(x)$  將對應一個長度為  $n - k$  的字 (向量) ,  $\mathbf{s}$  。現在 , 假設錯誤型為  $e(x)$  , 則  $w(x) = c(x) + e(x)$  ,  $s(x) \equiv e(x) \pmod{g(x)}$  , 所以  $e(x) = a(x)g(x) + s(x)$  , 而在  $\deg(e(x)) < n - k$  時 ,  $e(x) = s(x)$  , 這個概念在解碼時非常重要。

為了便於計算 , 定義一個適當的核對矩陣是非常必要的工作。令  $H$  為一  $n \times (n - k)$  的矩陣 , 其中  $H$  的第  $i$  列 , 恰為  $h_i(x) \equiv x^i \pmod{g(x)}$  所對應的  $n - k$  維向量 ,  $i = 1, 2, \dots, n$  。

**例.** 令  $C = \langle g(x) \rangle$  ,  $g(x) = 1 + x + x^3$  ,  $n = 7$  以及  $n - k = 3$  。

$$\begin{aligned} h_0(x) &\equiv 1 \pmod{g(x)} & h_4(x) &\equiv x^4 \equiv x + x^2 \pmod{g(x)} \\ h_1(x) &\equiv x \pmod{g(x)} & h_5(x) &\equiv x^5 \equiv 1 + x + x^2 \pmod{g(x)} \\ h_2(x) &\equiv x^2 \pmod{g(x)} & h_6(x) &\equiv x^6 \equiv 1 + x^2 \pmod{g(x)} \\ h_3(x) &\equiv x^3 \pmod{g(x)} \end{aligned}$$

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

現在 , 當  $w(x) = 1 + x^5 + x^6$  收到之後 ,  $w(x) \equiv 1 + x \pmod{g(x)}$  ; 以矩陣來看 , 則為  $(1, 0, 0, 0, 0, 1, 1)H = (1, 0, 0) + (1, 1, 1) + (1, 0, 1) = (1, 1, 0)$  , 它也是對應  $1 + x$  。

**命題 4.20.** 假如收到的字為  $w(x)$  , 則徵候多項式所對應的向量為  $\mathbf{w}H$  ,  $\mathbf{w}$  為  $w(x)$  所對應的向量。

**證明.** 直接乘。 ■

接下來看看如何解碼。由第二章的介紹，要解碼就需要知道每個餘集中加權最小的向量 Coset Leader，當  $\mathbf{w}H$  出現在該餘集中，選最小的加權向量為錯誤型 (Error Pattern) 即可更正，當然這個向量要有唯一性，否則無法正確地解碼。然而，是否每個餘集中都具有唯一的 Coset Leader 是可以由距離  $d$  來判斷，在這裡我們就假設 ”可以更正”，亦即唯一的 Coset Leader 不成問題。

於是，假設  $e(x)$  為錯誤型，而且對應於一個該餘集中加權最小的向量。如此一來  $x^i e(x) \pmod{1+x^n}$  也是一個 Coset Leader，只是所屬的餘集不同而已(?)。由於我們要判斷的是那一個  $e(x)$  是錯誤型，而它與  $s(x)$  有關，因此，以下的演算法可以派上用場。

#### 循環碼解碼演算法 (Decoding Algorithm for Linear Cyclic Codes)

1. 當  $w(x)$  收到時，計算  $s(x) \equiv w(x) \pmod{g(x)}$ 。
2.  $\forall i = 0, 1, 2, \dots, n-1$ ，計算  $s'_i(x) \equiv x^i s(x) \pmod{g(x)}$ 。
3. 令  $\mathbf{s}_j$  為對應於  $s'_j(x)$  的  $n-k$  維向量。
4. 當  $wt(\mathbf{s}_j) \leq \lfloor \frac{d-1}{2} \rfloor$  時， $e(x) \equiv x^{n-j} s'_j(x) \pmod{1+x^n}$  為所求的錯誤型。
5. Otherwise，做不出來。

**證明。** 因為  $s(x) \equiv e(x) \pmod{g(x)}$ ，所以  $x^i s(x) \equiv s'_i(x) \equiv x^i e(x) \pmod{g(x)}$ 。這表示當  $x^i e(x)$  所對應的向量如果加權不大於  $\lfloor \frac{d-1}{2} \rfloor$  時，必為一個 Coset Leader；再平移回去，就可以求得  $e(x) \equiv x^{n-i} s'_i(x) \pmod{1+x^n}$ 。爲了要使得上述得上述成立， $wt(\mathbf{s}_i) \leq \lfloor \frac{d-1}{2} \rfloor$ 。 ■

**例。**  $n = 15$ ， $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ ，” $d = 5$ ”。

收到  $\mathbf{w} = 110011100111000$ 。

$$w(x) = 1 + x + x^4 + x^5 + x^6 + x^9 + x^{10} + x^{11}。$$

$$s(x) = 1 + x + x^3 + x^4 + x^5 + x^6 + x^7。$$

$$s'_1(x)$$

⋮

$$s'_5(x) = x^5 s(x) \equiv 1 + x \pmod{g(x)}。$$

所以  $s'_5$  的加權為  $2 \leq \lfloor \frac{5-1}{2} \rfloor$ 。

$$e(x) \equiv x^{15-5} s'_5(x) \pmod{1+x^{15}} = x^{10} + x^{11}。$$

$$c(x) = e(x) + w(x) = 1 + x + x^4 + x^5 + x^6 + x^9 \longleftrightarrow 110011100100000。$$

(註) 這個演算法有它的缺點：因為它的錯誤型是利用  $x^i e(x)$  為 Coset Leader 的概念加上有  $i$  使得  $\deg(x^i e(x)) \leq n - k$  這個條件就可以完成。然而，對於不屬於這種型的  $e(x)$  就無能為力了。

**問題.** Decode 001000000110000 。

## 5. BCH 碼

BCH 代表三個人 Bose, Chaudhuri 及 Hocquengham，他們所建構出來的數碼具有很漂亮的結構。基本上，利用最基本的線性代數先找出核對矩陣，BCH 碼就因應地產生出來。

令  $\beta$  為有限體  $F = GF(2^r)$  中的一個 primitive 元素，也就是說  $\beta$  為乘法群  $\langle F^*, \cdot \rangle$  的生成元素 (Generator)。(  $\langle F^*, \cdot \rangle$  為循環群。) 所以， $F^* = \{1, \beta, \beta^2, \dots, \beta^{2^r-2}\}$ 。由於  $GF(2^r)$  的元素可以用次數小於  $r$  的多項式表示，也就可以用  $r$  維向量來表示，所以  $\beta^i$  本身可以看成是向量。

**命題 5.1.** 令  $\beta$  為  $F = GF(2^r)$  中的一個 primitive 元素， $H$  如下所示，則  $C = \{\mathbf{v} | \mathbf{v}H = \mathbf{0}\}$  為  $H_r$ 。

$$H = \begin{bmatrix} 1 \\ \beta \\ \beta^2 \\ \vdots \\ \beta^{2^r-2} \end{bmatrix}$$

**證明.**  $1, \beta, \beta^2, \dots, \beta^{2^r-2}$  為不等於  $\mathbf{0}$  的所有  $r$  維向量。 ■

**定義 5.2.** (2-error Correcting BCH Code)

令  $H$  為下所示的矩陣，其中  $\beta^i \in GF(2^r)$ ，則以  $H$  為核對矩陣的數碼為可以更正兩個錯誤的 BCH 碼。

$$H = \begin{bmatrix} 1 & 1 \\ \beta & \beta^3 \\ \beta^2 & \beta^6 \\ \vdots & \vdots \\ \beta^{2^r-2} & \beta^{3(2^r-2)} \end{bmatrix}$$

**命題 5.3.** 可更正兩個錯誤的 BCH 碼  $C$  是以  $g(x) = m_1(x)m_3(x)$  為生成多項式的循環碼，這裡  $m_i(x)$  為以  $\beta^i$  為零位的最小多項式 (minimal polynomial)。

**證明.** 令  $n = 2^r - 1$ ， $\mathbf{v} = v_0v_1 \cdots v_{n-1} \in C$ ，則  $\sum_{i=0}^{2^r-2} v_i(\beta^i\beta^{3i}) = \mathbf{0}$ 。以多項式表示，則  $h(x) = \sum_{i=0}^{2^r-2} v_ix^i = 0$  的根有  $\beta$  及  $\beta^3$ ， $h(x) \leftrightarrow \mathbf{v}$ 。因此， $m_1(x)|h(x)$  及  $m_3(x)|h(x)$ ，但是  $(m_1(x), m_3(x)) = 1$ ，所以  $m_1(x) \cdot m_3(x)|h(x)$ ，因此  $C \subseteq \langle g(x) \rangle$ ，又因為  $g(x) \in C$ ，所以  $C = \langle g(x) \rangle$ 。 ■

**命題 5.4.** 在上述的數碼中， $n = 2^r - 1$ ， $k = 2^r - 1 - 2r$ 。

**證明.** 我們證明  $\deg m_1(x) = \deg m_3(x)$ 。假設  $\deg m_1(x) = r$ ，所以  $s_1 = \left| \left\{ \beta^{2^i} \mid 0 \leq i \leq r-1 \right\} \right| = r$ ；現在考慮  $s_3 = \left| \left\{ (\beta^3)^{2^i} \mid 0 \leq i \leq r-1 \right\} \right|$ 。顯然  $s_3 \leq r$ 。但是，當  $(\beta^3)^{2^i} = (\beta^3)^{2^j}$  時， $3 \cdot 2^i \equiv 3 \cdot 2^j \pmod{2^r - 1}$ ， $3 \equiv 3 \cdot 2^{j-i} \pmod{2^r - 1}$ ，亦即  $j = i$  才有可能，所以  $s_3 = r$ 。 ■

在討論距離之前，我們回憶一下矩陣的行列式。

**引理 5.5.** 令  $\alpha_1, \alpha_2, \dots, \alpha_t$  為  $GF(2^r)$  中非零的元素，則下列矩陣  $A$  的行列式值為  $\prod_{1 \leq j < i \leq t} (\alpha_j + \alpha_i)$ 。

**證明.** 因為當  $\alpha_i = \alpha_j$  時 ( $i \neq j$ )， $\det A = 0$ ，所以  $\prod_{1 \leq j < i \leq t} (\alpha_j + \alpha_i) \mid \det A$ ，再利用兩邊的次數相同即可得證。 ■

**定理 5.6.** 令  $C = \langle g(x) \rangle$  而且  $(x + \beta^{m+1})(x + \beta^{m+2}) \cdots (x + \beta^{m+\delta-1}) \mid g(x)$  其中  $\beta$  為  $GF(2^r)$  的 primitive 元素，則  $d(C) \geq \delta$ 。

**證明.**

$$\text{令 } H' = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \beta^{m+1} & \beta^{m+2} & \cdots & \beta^{m+\delta-1} \\ (\beta^{m+1})^2 & (\beta^{m+2})^2 & \cdots & (\beta^{m+\delta-1})^2 \\ \vdots & \vdots & \ddots & \vdots \\ (\beta^{m+1})^{n-1} & (\beta^{m+2})^{n-1} & \cdots & (\beta^{m+\delta-1})^{n-1} \end{bmatrix}。$$

由於  $C$  中的任意元素與  $H'$  的每一行均垂直 (?) , 而任選  $H'$  的  $\delta - 1$  列, 它們都不會是相依的向量 ( 由引理 5.5 ) , 於是  $C$  中不為  $\mathbf{0}$  的碼字它的加權至少為  $\delta$  , 即  $d(C) \geq \delta$  。 ■

由於前面定義的 BCH 碼  $C = \langle m_1(x)m_3(x) \rangle$  , 所以  $(x + \beta)(x + \beta^2)(x + \beta^3)(x + \beta^4) \mid m_1(x)m_3(x)$  , 因此  $d(C) \geq 5$  , 也就自然可以更正兩個錯誤。為了方便起見, 以下稱呼  $\delta$  為設定距離 (Designed Distance) 。

接下來我們探討上述 BCH 碼的解碼方法。

令  $\mathbf{w}$  為收到的字,  $\mathbf{w} \leftrightarrow w(x)$ 。則  $\mathbf{s} = \mathbf{w}H = [w(\beta), w(\beta^3)] = [\mathbf{s}_1, \mathbf{s}_3]$  , 這裡的  $\mathbf{s}_1$  與  $\mathbf{s}_3$  分別是  $r$ -維向量。如果沒有任何錯誤產生,  $\mathbf{s} = \mathbf{0}$  , 這很容易判斷; 就是有一個錯誤產生, 例如  $e(x) = x^i$  , 則  $\mathbf{w}H = \mathbf{e}H = [e(\beta), e(\beta^3)] = [\beta^i, \beta^{3i}]$  , 於是  $\mathbf{s}_3 = \mathbf{s}_1^3$  ( 多項式 ) , 也很容易看出來; 最困難的部分是有兩個錯誤。

這個時候, 如果  $e(x) = x^i + x^j$  , 則  $\mathbf{w}H = [\mathbf{s}_1, \mathbf{s}_3] = [\beta^i + \beta^j, \beta^{3i} + \beta^{3j}]$  , 因此

$$\beta^i + \beta^j = s_1(x) \text{ and } \beta^{3i} + \beta^{3j} = s_3(x) \quad (1)$$

利用  $\beta^{3i} + \beta^{3j} = s_1(x)(s_1^2(x) + \beta^{i+j})$  , 推得

$$\frac{s_3(x)}{s_1(x)} + s_1^2(x) = \beta^{i+j} \quad (2)$$

另外,  $\beta^i, \beta^j$  為  $x^2 + (\beta^i + \beta^j)x + \beta^{i+j} = 0$  的兩根。

所以由 (1) 與 (2) ,  $\beta^i, \beta^j$  為  $y^2 + s_1(x)y + \left[ \frac{s_3(x)}{s_1(x)} + s_1^2(x) \right] = 0$  的兩根。

上式左邊也稱為是錯誤位置多項式 (Error-Locator Polynomial) 。

由於  $s_1(x)$  與  $s_3(x)$  都是  $r$ -維向量, 所以用  $GF(2^r)$  中的元素來表示就比較容易計算。

例.  $n = 15$  ,  $r = 4$  ,  $s_1(x) \leftrightarrow 0111 \leftrightarrow \beta^{11}$  ,

$$s_3(x) \leftrightarrow 1010 \leftrightarrow \beta^8, \frac{s_3}{s_1} + s_1^2 = \beta^8\beta^{-11} + \beta^{22} = \beta^{12} + \beta^7 = \beta^2,$$

所以, 錯誤位置多項式為  $y^2 + \beta^{11}y + \beta^2$  , 於是兩根為  $\beta^4$  與  $\beta^{13}$  , 這也告訴我們  $e(x) = x^4 + x^{13} \leftrightarrow 000010000000010$  。

## Decoding 2 Error-Correcting BCH Code (IMLD)

Algorithm (Generator Polynomial  $m_1(x)m_3(x)$  )

1. 收到  $\mathbf{w} \leftrightarrow w(x)$  , 求  $\mathbf{s}_1 = w(\beta)$  ,  $\mathbf{s}_3 = w(\beta^3)$  。
2. 假如  $\mathbf{s}_1 = \mathbf{s}_3 = \bar{0}$  , 則  $\mathbf{c} = \mathbf{w}$  。（沒有錯誤）
3. 假如  $\mathbf{s}_1 = \bar{0}$  ,  $\mathbf{s}_3 \neq \bar{0}$  則要求重送。(Retransmission)
4. 假如  $\mathbf{s}_1^3 = \mathbf{s}_3$  且  $\mathbf{s}_1 = \beta^i$  , 則  $e(x) = x^i$  。
5. 假如方程式  $y^2 + \mathbf{s}_1 y + \frac{\mathbf{s}_3}{\mathbf{s}_1} + \mathbf{s}_1^2 = 0$  有相異兩根  $\beta^i$  與  $\beta^j$  , 則  $e(x) = x^i + x^j$  。
6. 如果上述方程式在  $GF(2^r)$  中無法找到相異兩根 , 則可能錯誤多於兩個 , 要求重送。

證明. 前面以說明。 ■

問題. 令  $n = 15$  ,  $C = \langle 1 + x^4 + x^6 + x^7 + x^8 \rangle$  。

Decode (1) 110111011101100 , (2) 110011100100000 。

附錄.

$$GF(2^4) = Z_2[x]/\langle 1 + x + x^4 \rangle$$

|              |      |
|--------------|------|
| 1            | 1000 |
| $\beta$      | 0100 |
| $\beta^2$    | 0010 |
| $\beta^3$    | 0001 |
| $\beta^4$    | 1100 |
| $\beta^5$    | 0110 |
| $\beta^6$    | 0011 |
| $\beta^7$    | 1101 |
| $\beta^8$    | 1010 |
| $\beta^9$    | 0101 |
| $\beta^{10}$ | 1110 |
| $\beta^{11}$ | 0111 |
| $\beta^{12}$ | 1111 |
| $\beta^{13}$ | 1011 |
| $\beta^{14}$ | 1001 |

## 6. Reed-Solomon Codes

Reed-Solomon 碼為已知的數碼中應用最廣的一個；目前美國太空總署 (NASA) 及歐洲太空總署 (European Space Agency) 都在用它，而最重要的應用還有建構在光碟中的數碼也是屬於這一個群族的數碼。

RS-碼看起來很像 BCH-碼，不同的地方是在碼字中它的字母不再限定於  $GF(2)$  中的元素，它可以是  $GF(2^r)$  中的元素，也就是說它的碼字所對應的多項式是在  $GF(2^r)[x]$  裡面，當然，如果碼字的長度為  $n$ ，則所代表的多項式次數就會小於  $n$ 。

**定義 6.1.** ( $RS(2^r, \delta)$  碼)

一個  $RS(2^r, \delta)$  碼是一個佈於  $GF(2^r)$  的循環碼，它以  $g(x) = (x + \beta^{m+1})(x + \beta^{m+2}) \cdots (x + \beta^{m+\delta-1})$  為生成多項式，其中  $\beta$  為  $GF(2^r)$  的一個 primitive 元素。

**定理 6.2.** 令  $C$  為  $RS(2^r, \delta)$  碼，則以下的敘述為真。

- (1)  $n = 2^r - 1$ 。
- (2)  $k = 2^r - \delta$ 。
- (3)  $d(C) = \delta$ 。
- (4)  $|C| = 2^{rk}$ 。

**證明.** 除了 (3) 之外其餘很容易證明。由定理 5.6， $d(C) \geq \delta$ 。再利用定理 2.24  $d(C) \leq n - k + 1 = (2^r - 1) - (2^r - \delta) + 1 = \delta$ ，所以得證。 ■

所以由定義 2.25，我們知道  $RS(2^r, \delta)$  碼是一個最大距離分隔碼 (MDS-Code)，可以說是很好的數碼。另外，一個  $RS(2^r, \delta)$  碼也可以轉變成一個  $n = (2^r - 1) \times r$  的二元碼；這是因為  $RS(2^r, \delta)$  碼中，字母分別來自  $GF(2^r)$  中，所以，將這些字母用  $r$  維二元向量表示出來，就可以順利造出二元碼，而這種二元碼也具有非常好的性質。

**定義 6.3.** 群聚長度 (Burst Length)

令  $\mathbf{v} = (v_0, v_1, \dots, v_{n-1})$  為一二元向量，其中  $v_i$  與  $v_j$  分別為不等於 0 的第一個及最後一個分量，則群聚長度  $bl(\mathbf{v})$  定義為  $j - i + 1$ 。

**例.**  $\mathbf{v} = 1000100$ ， $bl(\mathbf{v}) = 5$ 。

**定義 6.4.** 循環群聚長度 (Cyclic Burst Length)

循環群聚長度  $cbl(\mathbf{v}) = \min \{ bl(\pi^k(\mathbf{v})) \mid k = 1, 2, \dots \}$  , 這裡的  $\pi^k(\mathbf{v})$  是  $\mathbf{v}$  的  $k$  次循環平移。

**例.**  $\mathbf{v} = 1000100$  ,  $cbl(\mathbf{v}) = 4$  。

如果以多項式的方式來看  $cbl(\mathbf{v})$  , 則是  $x^3(1+x^4) \bmod (1+x^7) = 1+x^3$ 。

**定義 6.5.** 可更正  $l$  個群聚錯誤的數碼  $C$  ( $l$  burst error-correcting code)

令  $H_{n \times (n-k)}$  為  $C$  的核對矩陣。如果對於所有群聚長度不大於  $l$  的  $n-k$  維向量都會屬於不同的餘集 (由  $H$  決定) , 則  $C$  可以更正  $l$  個群聚錯誤 , 也稱它是 "可更正  $l$  個群聚錯誤" 的數碼。

(註) 如果上述定義中的群聚錯誤改成循環群聚錯誤 , 則群聚長度不大於  $l$  要改成  $cbl \leq l$  。 ( $l$ -cyclic burst error-correcting code)

**命題 6.6.** 令  $C$  為  $l$ -cyclic 群聚錯誤更正碼 , 則在傳遞時所造成的錯誤型  $e(x) = x^k e'(x) \bmod (1+x^n)$  ,  $0 \leq k \leq n-1$  ,  $e'(x)$  為次數小於  $l$  的多項式。

**例.** 令  $g(x) = 1+x+x^2+x^3+x^6$  ,  $n=15$  ,  $C = \langle g(x) \rangle$  , 則  $C$  無法更正 3 個錯誤 , 但是可以更正長度為 3 的群聚錯誤。

**證明.** 具有 3 個錯誤以內的錯誤型一共有  $\binom{15}{3} + \binom{15}{2} + \binom{15}{1} + \binom{15}{0} = 576$  個 , 而餘集個數共有 64 個 ; 因此 , 要更正是不可能 , 然而 , 連續的型式之錯誤型有 61 個 , 機會較大。(另一部分列為作業。) ■

有了上述的概念 , 要解碼基本上是很容易 ; 當我們收到  $w(x)$  時 , 先計算  $s(x) \equiv w(x) \bmod g(x)$  , 然後在計算  $s_j(x) \equiv x^j s(x) \bmod g(x)$  , 直到有某一個  $j$  , 使得  $\deg(s_j(x)) \leq l-1$  , 則  $e(x) \equiv x^{n-j} s_j(x) \bmod (x^n + 1)$  為所求的錯誤型 (Error Pattern) 。

**例.** 同上例 , 假設收到的  $\mathbf{w} = 111100100001010$  , 則利用上述的概念 , 先求得  $s_4(x) \equiv 1+x^2 \bmod g(x)$  於是  $e(x) \equiv x^{15-4}(1+x^2) \bmod (x^{15} + 1) = x^{11} + x^{13}$  , 所以正確的  $\mathbf{v}$  為 111100100000000 。

**例.** 令  $n=15$  ,  $C = \langle 1+x^2+x^4+x^5 \rangle$  。

## 解碼的概念 (RS 碼)

因爲 RS 碼的字母不一定只有 0 與 1，而是  $GF(2^r)$  中的元素，所以當錯誤產生的時候，不但要找出錯誤的位置 (Location)，而且要知道錯的量有多大 (Magnitude)。爲了方便起見，錯誤的位置用錯誤數 (Error Location Number) 來表示：以  $\beta^{j-1}$  表示第  $j$  個位置，於是位置依次爲  $1, \beta, \beta^2, \dots, \beta^{j-1}$  ( $n = 2^r - 1$ )，至於錯誤數自然是  $GF(2^r)$  中的元素。

例. 在  $RS(8, 3)$  中，傳  $\mathbf{v} = \beta^3\beta^4 10000$ ，收到  $\mathbf{w} = \beta^3\beta^4\beta^5 0000$ ，則  $\mathbf{e} = 00\beta^4 0000$ ，因爲  $\beta^4 + \beta^5 = \beta^4(1 + \beta) = \beta^7 = 1$ ；這裡的錯誤位置爲  $\beta^2$ ，錯誤數爲  $\beta^4$ 。

接下來探討如何用一個演算法來解碼。

首先，令  $C = RS(2^r, \delta)$  碼的生成多項是爲  $g(x) = (x + \beta^{m+1})(x + \beta^{m+2}) \cdots (x + \beta^{m+\delta-1})$ ，其中  $\beta$  爲  $GF(2^r)$  的一個本元 (primitive 元素)；由於  $d(C) \geq \delta$ ，令  $t = \lfloor \frac{\delta-1}{2} \rfloor$ ， $a_1, a_2, \dots, a_e$  與  $b_1, b_2, \dots, b_e$  分別爲錯誤位置和錯誤數， $e \leq t$ 。爲了方便說明，令  $a_i = 0, e+1 \leq i \leq t$ 。這裡的假設主要來自錯誤位置的個數，並非一開始就知道，既然我們可以更正  $t$  個錯誤，我們直接假設錯誤的位置爲  $a_1, a_2, \dots, a_t$ ；如果沒有那麼多，在過程中也會被發現，如此才算是好的演算法。

現在，計算  $\delta-1$  個徵兆 (Syndromes)  $s_j(x) = w(\beta^j)$ ， $m+1 \leq j \leq m+\delta-1$ 。

由於  $w(x) = c(x) + e(x)$ ， $e(x) = \sum_{i=0}^{n-1} e_i x^i$ ，

$$s_j = w(\beta^j) = e(\beta^j) = \sum_{i=0}^{n-1} e_i (\beta^j)^i = \sum_{i=0}^{n-1} e_i (\beta^i)^j = \sum_{i=1}^t b_i a_i^j。$$

這裡，我們共有  $\delta-1$  個方程式，它們將被用來解最多  $2t$  個未知數  $a_1, a_2, \dots, a_t, b_1, b_2, \dots, b_t$ 。不過，這裡的方程式都不是線性方程式，所以比較複雜些。

令  $\sigma_r(x) = (a_1 + x)(a_2 + x) \cdots (a_t + x)$

$$= \sigma_0 + \sigma_1 x + \cdots + \sigma_{t-1} x^{t-1} + x^t \quad (*)$$

爲了解  $a_1, a_2, \dots, a_t$ ；先要求出  $\sigma_0, \sigma_1, \dots, \sigma_{t-1}$ 。(註) 一但有了 (\*)，就可以依次代入  $GF(2^r)$  中的元素來判斷那些元素是  $a_1, a_2, \dots, a_t$ 。現在把 (\*) 中的  $x$  以  $a_i$  代入，並且兩邊同乘  $\sum_{i=1}^t b_i a_i^j$ ，於是

$$\begin{aligned}
0 &= \left( \sum_{i=1}^t b_i a_i^j \right) \sigma_0 + \left( \sum_{i=1}^t b_i a_i^j \right) \sigma_1 a_i + \cdots + \left( \sum_{i=1}^t b_i a_i^j \right) \sigma_{t-1} a_i^{t-1} + \left( \sum_{i=1}^t b_i a_i^j \right) a_i^t \\
&= s_j \sigma_0 + s_{j+1} \sigma_1 + \cdots + s_{j+t-1} \sigma_{t-1} + s_{j+t}
\end{aligned}$$

所以  $s_{j+t} = s_j \sigma_0 + s_{j+1} \sigma_1 + \cdots + s_{j+t-1} \sigma_{t-1}$ 。

現在利用已知的  $s_{m+1}, s_{m+2}, \cdots, s_{m+2t}$ ，可以求得

$$\begin{bmatrix} s_{m+1} & s_{m+2} & \cdots & s_{m+t} \\ s_{m+2} & s_{m+3} & \cdots & s_{m+t+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m+t} & s_{m+t+1} & \cdots & s_{m+2t-1} \end{bmatrix} \begin{bmatrix} \sigma_0 \\ \sigma_1 \\ \vdots \\ \sigma_{t-1} \end{bmatrix} = \begin{bmatrix} s_{m+t+1} \\ s_{m+t+2} \\ \vdots \\ s_{m+2t} \end{bmatrix} \quad (**)$$

由於 (\*\*) 中最左邊的矩陣  $M$  可以寫成以下的型式，所以該矩陣可逆， $\sigma_0, \sigma_1, \cdots, \sigma_{t-1}$  也可以順利解出來，當然這是在有  $t$  個錯誤時的狀況，如果有些  $a_i = 0$ ，則可以由它的階數 (Rank) 看出。

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ a_1 & a_2 & \cdots & a_t \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{t-1} & a_2^{t-1} & \cdots & a_t^{t-1} \end{bmatrix} \begin{bmatrix} b_1 a_1^{m+1} & 0 & \cdots & 0 \\ 0 & b_2 a_2^{m+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & b_t a_t^{m+1} \end{bmatrix} \begin{bmatrix} 1 & a_1 & \cdots & a_1^{t-1} \\ 1 & a_2 & \cdots & a_2^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & a_t & \cdots & a_t^{t-1} \end{bmatrix}$$

有了  $\sigma_0, \sigma_1, \cdots, \sigma_{t-1}$  可以求出  $a_1, a_2, \cdots, a_t$ ；在利用

$$\begin{bmatrix} a_1^{m+1} & a_2^{m+1} & \cdots & a_t^{m+1} \\ a_1^{m+2} & a_2^{m+2} & \cdots & a_t^{m+2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{m+t} & a_2^{m+t} & \cdots & a_t^{m+t} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_t \end{bmatrix} = \begin{bmatrix} s_{m+1} \\ s_{m+2} \\ \vdots \\ s_{m+t} \end{bmatrix} \quad (***)$$

就可以算出  $b_1, b_2, \cdots, b_t$ 。

**演算法：**

1. 計算  $s_j(x) = w(\beta^j)$ ， $m+1 \leq j \leq m+2t$ 。

$$2. \text{ 求 } M' = \left[ \begin{array}{c|c} & \begin{matrix} s_{m+t+1} \\ s_{m+t+2} \\ \vdots \\ s_{m+2t} \end{matrix} \end{array} \right] \circ$$

3. 解 (\*\*)。

4. 求出  $a_1, a_2, \dots, a_t$  ( 用代入法, 或解方程式。)

5. 求  $b_1, b_2, \dots, b_t$  ( 用 (\*\*\*) )。

**例.**  $RS(2^4, 7)$  碼,  $g(x) = (1+x)(\beta+x)(\beta^2+x)(\beta^3+x)(\beta^4+x)(\beta^5+x)$ ,  $m = -1$ ,  $t = 3$ ,  $GF(2^4) = \mathbb{Z}_2[x]/\langle 1+x+x^4 \rangle$

$$w(x) = 1 + \beta^4 x + \beta x^3 + \beta^9 x^5 + x^6$$

$$s_0 = w(\beta^0) = \beta^7$$

$$s_1 = w(\beta^1) = 1$$

$$s_2 = w(\beta^2) = \beta^9$$

$$s_3 = w(\beta^3) = \beta^{12}$$

$$s_4 = w(\beta^4) = \beta^9$$

$$s_5 = w(\beta^5) = \beta^7$$

$$M' = \left[ \begin{array}{ccc|c} \beta^7 & 1 & \beta^9 & \beta^{12} \\ 1 & \beta^9 & \beta^{12} & \beta^9 \\ \beta^9 & \beta^{12} & \beta^9 & \beta^7 \end{array} \right] \leftrightarrow \left[ \begin{array}{ccc|c} \beta^7 & 1 & \beta^9 & \beta^{12} \\ 0 & \beta^{12} & \beta^7 & \beta^6 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

$$\text{所以, } wt(e) = 2, \left[ \begin{array}{cc} \beta^7 & 1 \\ 1 & \beta^9 \end{array} \right] \begin{bmatrix} \sigma_0 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \beta^9 \\ \beta^{12} \end{bmatrix}$$

$$\text{求得 } \sigma_0 = \beta^6, \sigma_1 = \beta^{10}$$

$$\sigma_A(x) = \beta^6 + \beta^{10}x + x^2, \text{ 於是 } a_1 = \beta^2, a_2 = \beta^4。$$

$$\text{再利用 } \begin{bmatrix} 1 & 1 \\ \beta^2 & \beta^4 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \beta^7 \\ 1 \end{bmatrix} \text{ 求得 } b_1 = \beta^2, b_2 = \beta^{12}。$$

$$\text{所以 } \mathbf{c} = \mathbf{w} + \mathbf{e} = 1\beta^4\beta^2\beta\beta^{12}\beta^9100000000。$$

**問題.** 同上例子的數碼，解碼下列兩個收到的字。

(1)  $0\beta^3\beta\beta^5\beta^3\beta^2\beta^6\beta^{10}\beta 0000000$  。

(2)  $\beta 0\beta^7 0\beta^{12}\beta^3\beta^3 10000000$  。

## 7. RS 碼 (續)

前面所提到向量與多項式的關係是：

$$\mathbf{v} = (v_0, v_1, v_2, \dots, v_{n-1}) \leftrightarrow f(x) = \sum_{i=0}^{n-1} v_i x^i ;$$

所以一個  $n$  維向量所對應的多項式次數為  $n-1$  次或是小一些。以下我們用函數的概念來看向量；令  $f: GF(2^r) \rightarrow \{0, 1\}$ ，則  $f$  可以用  $2^r$ -維向量表示出來，亦即  $\mathbf{v}_f = (f(0), f(1), f(\beta), \dots, f(\beta^{r-2}))$ 。

**例.**  $GF(2^3) = \mathbb{Z}_2[x]/\langle 1+x+x^3 \rangle$ ， $\beta$  為一 primitive 元素，則當  $\mathbf{v}_g = (\beta^4, 0, 1, \beta^2, 1, \beta, 0, 0)$  時， $g(x) = \beta^4 + \beta^2 x + \beta^3 x^2 + x^3$ ，這裡的函數  $g: GF(2^3) \rightarrow GF(2^3)$ 。

**定義 7.1.** 令  $p(x)$  與  $q(x)$  為兩個由  $S$  映至  $GF(2^r)$  的函數，其中  $S \subseteq GF(2^r)$ 。則  $p(x) = q(x)$  的充要條件為對於所有的  $\alpha \in S$ ， $p(\alpha) = q(\alpha)$ 。

現在考慮所有次數不大於  $k-1$  的多項式 (佈於  $GF(2^r)$ ) 所成的集合，則這個集合的元素可以看成向量，對於一般的加法和量積 (Scalar Product)，很容易核對出它形成一個向量空間， $\{1, x, x^2, \dots, x^{k-1}\}$  就是一組最明顯的基底。

**定理 7.2.** 令  $F_k$  ( $S \rightarrow GF(2^r)$  的向量形式) 為所有佈於  $GF(2^r)$  且次數不大於  $k-1$  的多項式所成的集合，則  $F_k$  是一個  $(n, k, n-k+1)$  MDS 碼，其中  $n \leq 2^r$ 。

**證明.** 令  $S \subseteq GF(2^r)$  且  $|S| = n$ 。

由已知，此線性碼  $F_k$  的維數為  $k$ ，現在證明  $d(C) = n - k + 1$ 。由於  $F_k$  中向量  $\mathbf{v}$ ，除了  $\bar{0}$  之外其它向量最多只有  $k-1$  個分量是 0，所以  $wt(\mathbf{v}) = n - (k+1)$ ，因此  $d(F_k) \geq n - k + 1$ ，但是由定理 2.24， $d(F_k) \leq n - k + 1$ ，所以得證。 ■

上述定理所得到的數碼已經不錯，然而，不一定是循環碼，這與希望建構的碼仍有一點距離。

**定義 7.3.** (nth roots of unity)

集合  $S = \{\alpha \in F = GF(2^r) \mid \alpha^n = 1\}$  為 1 的  $n$  次方根。

**定義 7.4.** 由於  $\alpha \in GF(2^r)$  而且  $\alpha^n = 1$  , 所以  $n | 2^r - 1$  , 以下是一個  $GF(2^4)$  的例子 , 其中  $GF(2^4) = Z_2[x]/\langle 1+x+x^4 \rangle$  ,  $\beta$  為一個 primitive 元素。於是  $\{1, \beta^3, \beta^6, \beta^9, \beta^{12}\}$  中的每一個元素均為 1 的 5 次方根 ; 而 3 次方根有  $\{1, \beta^5, \beta^{10}\}$  。

**定理 7.5.** 令  $S = \{\alpha \in F = GF(2^r) | \alpha^n = 1\}$  而且  $p(x)$  與  $q(x)$  為表示由  $S$  映至  $GF(2^r)$  的兩個函數。則  $p(x)$  ,  $q(x)$  代表同一函數的充要條件為  $p(x) \equiv q(x) \pmod{(1+x^n)}$  。

**證明.**

( $\Leftarrow$ ) 假設  $\deg(p(x)) \leq \deg(q(x))$  , 於是  $q(x) = h(x)(1+x^n) + p(x)$  ,  $\deg(p(x)) < n$  。  $\forall \alpha \in S$  ,  $q(\alpha) = h(\alpha)(1+\alpha^n) + p(\alpha)$  , 所以  $p(x)$  、  $q(x)$  代表同一函數。

( $\Rightarrow$ ) 令  $\alpha \in S$  , 則  $p(\alpha) - q(\alpha) = 0$  , 所以  $(1+x^n) | p(x) - q(x)$  。 ■

**定理 7.6.** 令  $S = \{\alpha \in F = GF(2^r) | \alpha^n = 1\}$  , 則  $F_k$  是一個佈於  $GF(2^r)$  的  $(n, k, n-k+1)$  循環碼。

**證明.** 由定理 7.2 , 我們只需要證明循環的性質即可。

令  $\mathbf{v}_p = (P(1), P(\beta), \dots, P(\beta^{n-1})) \in C$  ,  $S = \{1, \beta, \beta^2, \dots, \beta^{n-1}\}$  。 令  $p'(x) = p(\beta x)$  , 顯然  $\deg(p'(x)) \leq k-1$  , 所以  $\mathbf{v}_{p'} \in C$  ; 但是  $\mathbf{v}_{p'} = (P(\beta), P(\beta^2), \dots, P(\beta^{n-1}), P(1))$  , 所以任何的循環平移都會是碼字 , 定理得證。 ■

**例.**  $p(x) = \beta^4 + \beta^2 x + \beta^3 x^2 + x^3$  ( 多項式  $1+x+x^3$  )

則  $\mathbf{v}_p = (0, 1, \beta^2, 1, \beta, 0, 0)$  ,

$\mathbf{v}_{p'} = (1, \beta^2, 1, \beta, 0, 0, 0) \leftrightarrow (\beta^4 + x)(\beta^5 + x)(\beta^6 + x)\beta^3$  。

**定義 7.7.** 變換 (Transform)

令  $V(x) = V_0 + V_1 x + V_2 x^2 + \dots + V_{n-1} x^{n-1}$  為一多項式 , 則  $v(x) = \sum_{i=0}^{n-1} v_i x^i$

為  $V(x)$  的一個變換若且唯若  $V(\beta^j) = \sum_{i=0}^{n-1} V_i \beta^{ij} = v_j$  ,  $j = 0, 1, 2, \dots, n-1$  。

以矩陣的方式來看 , 令  $A = (a_{ij}) = (\beta^{ij})$  , 則  $(V_0, V_1, \dots, V_{n-1})A = (v_0, v_1, \dots, v_{n-1})$  所以  $(V_0, V_1, \dots, V_{n-1}) = (v_0, v_1, \dots, v_{n-1})A^{-1}$  。

(註) 由於  $\beta$  是 1 的 primitive  $n$  次方根 , 所以  $A$  可逆 , 一般  $A$  也稱為是有限的富氏變換 (Finite Fourier Transform) 。

**定理 7.8.** 如果  $\beta$  是 1 的 primitive  $n$  次方根,  $v(x) = \sum_{i=0}^{n-1} v_i x^i$  為  $V(x) = \sum_{i=0}^{n-1} V_i x^i$ , 的富氏變換, 則  $V_j = v(\beta^{-j})$ 。

**證明.**

$$\begin{aligned} v(\beta^{-j}) &= \sum_{i=0}^{n-1} v_i \beta^{-ij} = \sum_{i=0}^{n-1} \left( \sum_{k=0}^{n-1} V_k \beta^{ki} \right) \beta^{-ij} \\ &= \sum_{k=0}^{n-1} V_k \left( \sum_{i=0}^{n-1} \beta^{(k-j)i} \right) \end{aligned} \quad (2)$$

由於當  $k = j$  時,  $\sum_{i=0}^{n-1} \beta^{(k-j)i} = n \pmod{2}$ , 而  $n \nmid 2^r - 1$ , 所以是奇數, 亦即  $\sum_{i=0}^{n-1} \beta^{(k-j)i} = 1$ , 另一方面, 當  $k - j \neq 0$  時,  $\beta^{k-j}$  為  $x^n + 1 = 0$  的根,  $\sum_{i=0}^{n-1} \beta^{(k-j)i}$  為  $n$  個根的和, 所以  $\sum_{i=0}^{n-1} \beta^{(k-j)i} = 0$ , 於是  $(2) = V_j$  ( $k = j$ )。 ■

**定理 7.9.** 令  $S = \{\alpha \in F = GF(2^r) \mid \alpha^n = 1\}$ 。則所有由次數小於  $n - \delta + 1$  對應出來的函數 (向量形式) 等價於一個  $g(x) = (1+x)(\beta+x) \cdots (\beta^{\delta-1}+x)$  為生成多項式的循環碼 MDS 碼, 這裡的  $\beta$  是  $GF(2^r)$  中的一個 1 的 primitive  $n$  次方根。

**證明.** 令  $C(x)$  為一函數, 而它的向量形式對應的多項式為  $c(x)$ 。由於  $C(x) = \sum_{i=0}^{n-1} C_i x^i = \sum_{i=0}^{n-1} c(\beta^{-i}) x^i = \sum_{i=0}^{n-1} c(\beta^{n-i}) x^i$ , 現在, 當  $g(x) \mid c(x)$  時,  $\forall i = n - \delta + 1, n - \delta + 2, \dots, n - 1$ ,  $c(\beta^{n-i}) = 0$  所以  $\deg(C(x)) < n - \delta + 1$ 。 ■

(\*) 上面得到的結論提供了以 Transform 表示的數碼, 也因此, 生成矩陣也會不一樣。

**例.** 令  $RS(2^3, 5)$  的生成多項式為  $g(x) = (1+x)(\beta^2+x)(\beta^3+x)$

$$= \beta^6 + \beta^5 x + \beta^5 x^2 + \beta^2 x^3 + x^4 \leftrightarrow (\beta^6, \beta^5, \beta^5, \beta^2, 1, 0, 0),$$

$$g(x) \text{ 的變換 } G(x) = \sum_{i=0}^6 g(\beta^{7-i}) x^i, \beta^7 = \beta^0,$$

$$\begin{aligned} G(x) &= g(\beta^{7-1})x + g(\beta^{7-2})x^2 + g(\beta^{7-3})x^3 \\ &= \beta^4 x + \beta x^2 + x^3 = x(\beta^4 + \beta x + x^2) \end{aligned} \quad (3)$$

於是, 向量形成  $(G(\beta^0), G(\beta^1), \dots, G(\beta^6)) = (\beta^6, \beta^5, \beta^5, \beta^2, 1, 0, 0)$ , 於是這個  $RS(2^3, 5)$  碼可以看成是由多項式次數一次, 二次及三次所對應的函數所組成。(或是  $x p(x) \pmod{(1+x^7)}$ ,  $\deg(p(x)) < 3$ )。

由於  $\{x, x^2, x^3\}$  是一組基底，而且

$$x \leftrightarrow (1, \beta, \beta^2, \beta^3, \beta^4, \beta^5, \beta^6)$$

$$x^2 \leftrightarrow (1, \beta^2, \beta^4, \beta^6, \beta, \beta^3, \beta^5)$$

$$x^3 \leftrightarrow (1, \beta^3, \beta^6, \beta^2, \beta^5, \beta, \beta^4)$$

所以當  $G(x) = \beta^4 x + \beta x^2 + x^3$ ，它的向量形式就是

$$(\beta^4, \beta, 1) \begin{bmatrix} 1 & \beta & \beta^2 & \beta^3 & \beta^4 & \beta^5 & \beta^6 \\ 1 & \beta^2 & \beta^4 & \beta^6 & \beta & \beta^3 & \beta^5 \\ 1 & \beta^3 & \beta^6 & \beta^2 & \beta^5 & \beta & \beta^4 \end{bmatrix}$$

解碼。

概念：函數形式的次數較低，複雜度較小。

傳  $\mathbf{c}$ ，收到  $\mathbf{w}$ ， $\Rightarrow w(x) = c(x) + e(x)$



(Error Polynomial)

經變換之後  $W(x) = C(x) + E(x)$

$$W(x) = \sum_i w(\beta^{n-i})x^i = \sum_i [c(\beta^{n-i}) + e(\beta^{n-i})]x^i$$

因為  $g(\beta^i) = 0$ ， $i = m+1, m+2, \dots, m+\delta-1$

所以  $c(\beta^i) = 0$ ，亦即  $E_{n-i} = e(\beta^i) = w(\beta^i) = s_i$  for  $n-i = m+1, m+2, \dots, m+\delta-1$ 。由於  $E(x) = \sum_{i=0}^{n+\delta-1} E_i x^i$ ，所以，除了前面的  $E_0, E_1, \dots, E_m$  尚且不知外，其它的已經可以由 Syndrome  $s_{n-i}$  求得。接下來討論如何求出這些未知的係數。

首先，回憶一下碼的解碼；如果  $\beta^k$  是一個錯誤位置，則  $\sigma(\beta^k) = 0$ 。  
((註)  $\beta^k$  是一位錯誤位置若且唯若  $e_k \neq 0$ 。)

另外，因為  $E(\beta^k) = e_k$ ，所以  $\sigma(\beta^k)E(\beta^k) = 0$ ， $k = 0, 1, \dots, n-1$ 。這表示  $1+x^n \mid \sigma(x)E(x)$ ，亦即  $\sigma(x)E(x) \equiv 0 \pmod{1+x^n}$ ，

$$\left( \sum_{i=0}^t \sigma_i x^i \right) \left( \sum_k E_k x^k \right) \equiv 0 \pmod{1+x^n} \quad (4)$$

$$\text{由 (4), } \sigma_t E_k + \sigma_{t-1} E_{k+1} + \dots + \sigma_0 E_{k+t} = 0 \text{ ( } x^{t+k} \text{ 的係數 )} \quad (5)$$

由於連續的  $\delta - 1$  個  $E_k$  為已知，所以先用它們來求出  $\sigma_0, \sigma_1, \dots, \sigma_t$ ；再不斷的求出所有的  $E_k$ 。

例.  $RS(2^3, 5)$  碼， $m = -1$ ， $t = 2$ ， $g(x) = (1 + x)(\beta + x)(\beta^2 + x)(\beta^3 + x)$ ， $\mathbf{w} = (\beta^6, \beta, \beta^5, \beta^2, 1, 0, \beta^2)$ ，

$$E_0 = w(\beta^0) = 1, E_6 = w(\beta) = \beta^3, E_5 = w(\beta^2) = \beta^3, E_4 = w(\beta^3) = 1$$

已知  $E_0, E_6, E_5, E_4$ ，我們要求  $E_3, E_2$  及  $E_1$

(1) 先求出  $\sigma_0, \sigma_1$ ，(  $\sigma_A(x) = x^2 + \sigma_1 x + \sigma_0$ ，利用上一章的演算法。)

$$\sigma_0 = 1, \sigma_1 = \beta^5$$

於是，

$$(2) E_3 = \beta^5 E_4 + E_5 = \beta^5 + \beta^3 = \beta^2$$

$$E_2 = \beta^5 E_3 + E_4 = \beta^5 + \beta^2 + 1 = 0$$

$$E_1 = \beta^5 E_2 + E_3 = 0 + \beta^2 = \beta^2$$

$$(3) E(x) = \sum E_k x^k, (E_0, E_1, \dots, E_6) = (1, \beta^2, 0, \beta^2, 1, \beta^3, \beta^3)。$$

$$(4) \mathbf{e} = (E(\beta^0), E(\beta), \dots, E(\beta^6)) = (0, \beta^6, 0, 0, 0, 0, \beta^2)。$$

$$(5) \mathbf{c} = \beta^6 \beta^5 \beta^5 \beta^2 100。$$

問題. 利用這一章所介紹的方法來解碼， $\mathbf{w}$  如上一章的問題。

## 8. 群聚錯誤更正碼 (Burst Error-Correcting Codes)

前面已經介紹過所謂的群聚錯誤 (Burst Error) 以及循環群聚錯誤 (Cyclic Burst Error) , 在這一章中我們介紹這種形式的數碼。

**定理 8.1.** 如果  $C$  是可以更正  $l$  個群聚錯誤的  $(n, k)$  碼 , 則  $l \leq n - k$  。

**證明.** 令  $\mathbf{e} = e_1 e_2 \cdots e_n$  為錯誤型 , 現在考慮  $e_i = 0$  或  $1$  ,  $1 \leq i \leq l$  ,  $e_k = 0$  ,  $k > l$  的所有錯誤型 , 一共個  $2^l$  ; 因為  $C$  可以更正  $l$  個群聚錯誤 , 所以這  $2^l$  個錯誤型在解碼時應該要落在  $2^l$  個不同的餘集中 , ( 反應出  $2^l$  個 Syndromes ) , 因此  $C$  的維數  $k$  最多只有  $n - l$  , 亦即  $l \leq n - k$  。 ■

顯然 , 如果  $l$  能夠接近  $n - k$  , 這樣的更正能力已經相當不錯。接下來看二元化 RS 碼的更正能力。

**定理 8.2.** 令  $C$  為一個  $RS(2^r, 2t + 1)$  碼 , 而  $\hat{C}$  則是由  $C$  中的碼字轉換出來的二元碼字 , ( 所以  $\hat{C}$  的長度為  $r(2^r - 1)$  ) 。則  $\hat{C}$  是可以更正  $l$  個群聚錯誤的數碼 , 其中  $l \leq n(t - 1) + 1$  。

**證明.** 令  $\hat{\mathbf{e}}$  為  $\hat{C}$  中傳遞  $\hat{\mathbf{v}}$  所造成的一個錯誤型 , 於是  $\hat{\mathbf{w}} = \hat{\mathbf{v}} + \hat{\mathbf{e}}$  是收到的字 , 現在把  $\hat{\mathbf{w}}$  對應回去  $\bar{\mathbf{w}}$  , 最多產生  $t$  個錯誤 (?) , 這是可以更正的錯誤數 , 所以可以求出  $C$  中的  $\mathbf{v}$  , 再對應至  $\hat{\mathbf{v}}$  即為所求。 ■

**例.**  $RS(2^3, 5)$  可以更正 4 個群聚錯誤。 (  $r = 3$  ,  $t = 2$  )

為了製造可以更正較多的群聚錯誤 , 我們介紹一種把碼字拆開來傳遞的概念 (Interleaving) :

令  $\bar{v}_1, \bar{v}_2, \cdots, \bar{v}_s$  為  $s$  個長度為  $n$  的碼字 ;  $\bar{v}_i = (v_{i,1}, v_{i,2}, \cdots, v_{i,n})$  ,  $i = 1, 2, \cdots, s$  , 傳遞字母的順序換成

$$v_{1,1}, v_{2,1}, \cdots, v_{s,1} ; v_{1,2}, v_{2,2}, \cdots, v_{s,2} ; \cdots ; v_{1,n}, v_{2,n}, \cdots, v_{s,n} ;$$

則這樣的傳地方是稱為是深度  $s$  的輪替式 (Interleaving to depth  $s$ ) 傳遞。

這個概念可以想像成將個碼字放成  $s$  個碼字放成  $s \times n$  矩陣的  $s$  列 , 傳遞時 , 依行的順序 , 先傳第一行 , 再來第二行 ,  $\cdots$  。

例.  $\bar{v}_1 = 100110$   
 $\bar{v}_2 = 010101$   
 $\bar{v}_3 = 111000$   
 $\bar{v}_4 = 010101$   
 $\bar{v}_5 = 100110$   
 $\bar{v}_6 = 111000$

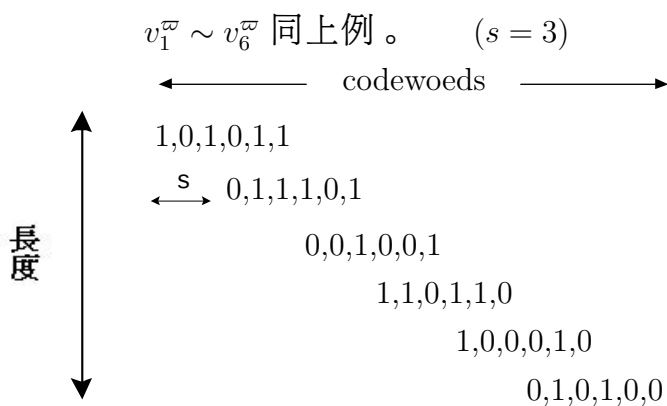
傳成 : 101011 | 011101 | 001001 | 110110 | 100010 | 010100  
(depth 6)

**定理 8.3.** 假設一個碼字的傳送最多只會受到一個群聚錯誤型所影響，使用的數碼可以更正  $l$  個群聚錯誤，而且使用深度為  $s$  的輪替傳送，則可以更正的群聚錯誤數可以是  $sl$ 。

**證明.** 這個可以由傳送的形式看出來； $sl$  個群聚錯誤對應於原來的數碼，對於每個碼字而言，最多造  $l$  個群聚錯誤，每個碼字因此可以正確的解碼。 ■

一次傳  $s$  個碼字有它的優點，但是缺點也有：在沒有 ”擺好”  $s$  個碼字的全部訊息之前無法傳遞。現在，我們介紹一種不同的傳遞方法。

例.



一行一行輪流傳送，空白的部分補上 ”\*” 或 ”0”。

1\*\*\*\*\*0\*\*\*\*\*1\*\*\*\*\*00\*\*\*\*\*11\*\*\*\*\*...

以上的傳遞方法稱為是  $s$ -frame delay interleaved (SFDI) 的傳遞法。在這樣的形勢下所能更正的群聚錯誤就更多了。

**定理 8.4.** 假設每一個碼字最多只受到一個群聚錯誤型所影響，則當  $C$  是可以更正  $l$  個群聚錯誤的數碼以及傳遞形式採 SFDI 進行，則這個數碼可以更正  $l(sn+1)$  個錯誤。

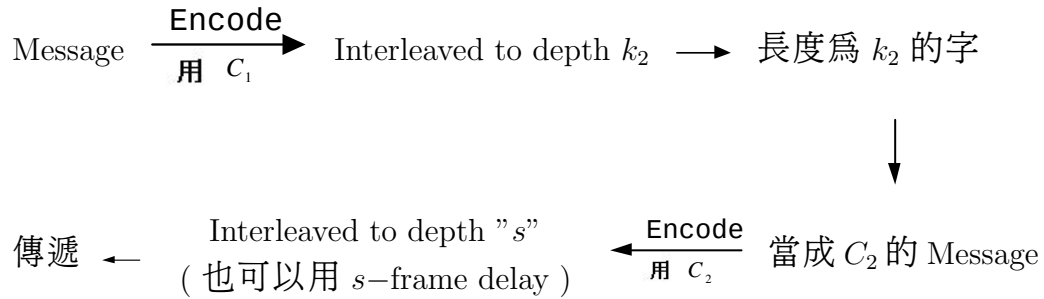
**證明.** 具  $l(sn+1)$  個群聚錯誤的形式對於每一個碼字最多產生  $l$  個群聚錯誤，因此都可以更正。 ■

**問題.** 仔細證明定理 8.4 (Assignment 5(1))。

在實際的使用上，我們在編碼 (Encode) 時，常常用兩個數碼  $C_1$  及  $C_2$ 。

**定義 8.5.** 交替式編碼 (Cross-Interleaving)

令  $C_1$  及  $C_2$  分別為  $(n_1, k_1, d_1)$  及  $(n_2, k_2, d_2)$  線性碼。



**定理 8.6.** 利用上述方法，如果每個碼字只受到一個群聚錯誤型影響，則當  $s \leq d_1 - 1$  時，可更正  $s(d_2 - 1)$  個群聚錯誤。

**例.** 令  $C_1$  與  $C_2$  的生成矩陣分別如下：

$$G_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$C_1 : (8, 4, 4)$ -碼       $C_2 : (6, 3, 3)$ -碼

Messages :  $m_1 : 1000$  ,  $m_2 : 1100$  ,  $m_3 : 1010$

Encoding : Cross interleaving  $C_1$  with  $C_2$  , depth 3.

$$c_1 = m_1 G_1 = 10001110$$

$$c_2 = m_2 G_1 = 11000011$$

$$c_3 = m_3 G_1 = 10100101$$

111 , 010 , 001 , 000 , 100 , 101 , 110 , 011



use  $G_2$  to encode

111000 , 010101 , 001011 , 000000 , 100110 , 101101 , 110011 , 011110



depth 3

111000 | 000000 | 110011

010101 | 100110 | 011110

001011 | 101101

← 下一組的第一個碼字

100110101010001011011000001011010001...

根據定理 8.4 , 6 個群聚錯誤可以被更正。現在假設最前面的 6 個字母有錯 , 收到的是 :

011001101010001011...



換回原來形式

001000 , 100101 , 111011



錯誤的位置 ( 可能 )

利用  $C_2$  可以發現有錯誤。

接下來將這些字母全部加註記號 (Flagged) , 亦即

\*\*\*\*\*...

如果這 18 個位置是僅有的錯誤 , 就可以將它們拉回  $C_1$  去看 , 於是

$$c_1 = * * * 01110$$

$$c_2 = * * * 00011$$

$$c_3 = * * * 00101$$

更正的部分可以借用  $C_1$  的性質 (?)。

## 9. RS 碼的應用

首先介紹擦拭掉 (Erasure) 的概念。由於 RS 碼或是其它非二元碼在產生錯誤時，除了知位置之外還要知道錯誤的量有多少，才能正確地更正它，但是，就整體而言，能夠知道一些錯誤的位置，卻有助於解碼。

**定義 9.1.** (Erasure)

一個 "Erasure" 是一個不知道錯誤量 (Magnitude) 的錯誤位置。

**例 1.**  $RS(2^2, 2)$  碼  $C$ ，生成矩陣為

$$G = \begin{bmatrix} \beta & 1 & 0 \\ 0 & \beta & 1 \end{bmatrix}$$

在  $GF(2^2)$  中  $0, 1, \beta, \beta^2$  分別為 00, 10, 01, 11。

|           |           |   |   |   |   |   |   |   |           |           |           |   |   |   |   |   |   |
|-----------|-----------|---|---|---|---|---|---|---|-----------|-----------|-----------|---|---|---|---|---|---|
| 0         | 0         | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0         | $\beta^2$ | $\beta$   | 0 | 0 | 1 | 1 | 0 | 1 |
| $\beta$   | 1         | 0 | 0 | 1 | 1 | 0 | 0 | 0 | $\beta$   | $\beta$   | $\beta$   | 0 | 1 | 0 | 1 | 0 | 1 |
| $\beta^2$ | $\beta$   | 0 | 1 | 1 | 0 | 1 | 0 | 0 | $\beta^2$ | 1         | $\beta$   | 1 | 1 | 1 | 0 | 0 | 1 |
| 1         | $\beta^2$ | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1         | 0         | $\beta$   | 1 | 0 | 0 | 0 | 0 | 1 |
| 0         | $\beta$   | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0         | 1         | $\beta^2$ | 0 | 0 | 1 | 0 | 1 | 1 |
| $\beta$   | $\beta^2$ | 1 | 0 | 1 | 1 | 1 | 1 | 0 | $\beta$   | 0         | $\beta^2$ | 0 | 1 | 0 | 0 | 1 | 1 |
| $\beta^2$ | 0         | 1 | 1 | 1 | 0 | 0 | 1 | 0 | $\beta^2$ | $\beta^2$ | $\beta^2$ | 1 | 1 | 1 | 1 | 1 | 1 |
| 1         | 1         | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1         | $\beta$   | $\beta^2$ | 1 | 0 | 0 | 1 | 1 | 1 |

|   |   |   |          |   |   |   |   |   |
|---|---|---|----------|---|---|---|---|---|
| 0 | 0 | 0 | 0        | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1        | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0        | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1        | 1 | 0 | 0 | 0 | 0 |
|   |   |   | $\vdots$ |   |   |   |   |   |
| 1 | 0 | 1 | 0        | 1 | 1 | 1 | 1 | 0 |

收到 011 100 000

錯誤位置 ( $RS(2^2, 2)$  碼) 為  $\beta^1, \beta^1$  為 Erasure, 所以  $\alpha_1 = \beta$ ; 爲了方便起見, 可以令要解碼的  $\mathbf{w} = (\beta, 0, 0)$ 。

**定理 9.2.** 令  $C$  爲  $RS(2^r, \delta)$  碼且  $\mathbf{w}$  爲傳送收到的字。如果再傳地中發生  $\varepsilon$  個被擦拭掉的位置及  $e$  個錯誤, 則  $\mathbf{w}$  可以被正確解碼條件是  $2e + \varepsilon \leq \delta - 1$ 。

**證明.** 令  $B$  爲  $\varepsilon$  個 Erasures 所成的集合,  $A$  爲所有錯誤位置所成的集合, 則  $A - B$  爲不是 Erasure 的錯誤位置, 所以

$$\sigma_B(x) = \prod_{\beta^i \in B} (x + \beta^i) = B_0 + B_1x + \cdots + B_{\varepsilon-1}x^{\varepsilon-1} + x^\varepsilon,$$

$$\sigma_A(x) = \sigma_B(x)\sigma_{A-B}(x) \circ$$

$$\text{令 } \sigma_{A-B}(x) = A_0 + A_1x + \cdots + A_{e-1}x^{e-1} + x^e,$$

$$\sigma_A(x) = \prod_{i=1}^{\varepsilon+e} (x + a_i), \text{ 以及}$$

$$\prod_{i=1}^{\varepsilon+e} (x + a_i) = (B_0 + B_1x + \cdots + B_{\varepsilon-1}x^{\varepsilon-1} + x^\varepsilon)(A_0 + A_1x + \cdots + A_{e-1}x^{e-1} + x^e) (*)$$

(\*) 的兩邊同乘  $\sum_{i=1}^{\varepsilon+e} b_i a_i^j = s_j, m+1 \leq j \leq m+\delta-1$ , 並且代入  $a_i$ , 可以得到

$$\begin{aligned} & (B_0s_j + B_1s_{j+1} + \cdots + B_{\varepsilon-1}s_{j+\varepsilon-1} + s_{j+\varepsilon})A_0 + \\ & (B_0s_{j+1} + B_1s_{j+2} + \cdots + B_{\varepsilon-1}s_{j+\varepsilon} + s_{j+\varepsilon+1})A_1 + \cdots + \\ & (B_0s_{j+e} + B_1s_{j+e+1} + \cdots + B_{\varepsilon-1}s_{j+e+\varepsilon-1} + s_{j+e+\varepsilon}) = 0. \end{aligned} \quad (**)$$

定義  $s_j^* = B_0s_j + B_1s_{j+1} + \cdots + B_{\varepsilon-1}s_{j+\varepsilon-1} + s_{j+\varepsilon}, m+1 \leq j \leq m+\delta-1$

因爲  $B_0, B_1, \cdots, B_{\varepsilon-1}, s_{m+1}, s_{m+2}, \cdots, s_{m+\delta-1}$  爲已知, 所以

$s_{m+1}^*, s_{m+2}^*, \cdots, s_{m+\delta-1-\varepsilon}^*$  可以求出來。 ( $\delta-1-\varepsilon \geq 2e$ )

$$\text{由 } (**) \begin{cases} s_{m+e+1}^* = A_0s_{m+1}^* + \cdots + A_{e-1}s_{m+e}^* \\ s_{m+2e}^* = A_0s_{m+e}^* + \cdots + A_{e-1}s_{m+2e-1}^* \end{cases}$$

解碼方程式求得  $A_0, A_1, \dots, A_{e-1}$  ; 於是錯誤的位置全部決定出來 ; 最後再利用 RS 碼的解碼演算法求出錯誤量即可。 ■

接下來我們介紹縮短碼 (Shorten Code) 。當一個數碼建構好之後 , 我們可選擇一些位置 , 然後把在那些位置都是 "0" 的碼字選出來後再去掉這些位置 , 如此即可製造出一個縮短的數碼 , 它在距離方面不會變小。

**例 2.** (Shorten RS-code)

特例 1 中的  $RS(2^2, 2)$  碼的最後一個位置去掉 "0" , 就得到  $C(1) = \{00, \beta^1, \beta^2, 1\beta^2\}$  。這裡  $C(1)$  中的 "1" 代表最後一個位置 ; 如果是最後的  $s$  個位置 , 則以  $C(s)$  表示。

**例 3.** 令  $C$  為  $RS(2^8, 5)$  碼 ,  $n = 2^8 - 1$  ,  $k = 2^8 - 5$  以及  $d = 5$  。

現在考慮  $C(227)$  , 碼字的長度為  $225-227=28$  , 由於這些碼字仍會形成 RS 碼的子碼 , 所以  $d = 5$  ,  $k = 24$  , 同理  $C(223)$  為一個  $(32, 28, 5)$  碼。

## CD 的設計 (Design of Compact Disc)

### 編碼.

1. 螺旋紋路 (Spiral Track) 。
2. 溝槽深度有兩種不同的深度 : 深度變化處為 1 , 其它是 0 , 連續的 "0" 最少 2 個最多 10 個。
3. 音階一共有  $2^{16}$  個 , 亦即每個音階用 16 維的二元向量表示。
4. 16 維向量可以寫成  $(\alpha^\omega, \beta^\omega)$  ,  $\alpha^\omega, \beta^\omega$  為  $GF(2^8)$  中的元素所表成的 8 維向量。
5. 立體音樂 (Stereo) 一次輸入兩個音 , 一左一右 , 所以每傳遞一次 (Tick), 就出現 4 個連續的  $GF(2^8)$  之元素  $m_{4t}, m_{4t+1}, m_{4t+2}, m_{4t+3}$  。
6. 6 個 Ticks 合成一訊息 , 長度 24 :  $m_{24t}, m_{24t+1}, \dots, m_{24t+22}, m_{24t+23}$  。
7. 令  $C_1 = C(227) : (28, 24, 5)$ -RS 碼。  
將長度為 24 的訊息編碼成長度為 28 的碼字。
8. 將  $C_1$  的碼字用 4-frame delay interleaved 起來。(參考 p.56) , 此時長度為 28 。
9. 令  $\mathbf{v}_t = (v_{1,t}, v_{2,t+4}, v_{3,t+8}, \dots, v_{28,t+108})$  。

10. 傳送的表格中第  $t$  行中，有下列的元素：

$$\mathbf{c}_t = (v_{1,t}, v_{2,t}, v_{3,t}, \dots, v_{28,t})。$$

11. 把  $\mathbf{c}_t$  用 (32, 28, 5) RS 碼  $C(223)$  編碼成  $\mathbf{v}_t$ ，然後再加入一個用來控制與顯示的 "byte"，得到一個長度為 33 的碼字  $\mathbf{v}$ 。
12. (Eight to Fourteen Modulation) 在  $\mathbf{v}$  中的每一個字母用長度為 17 的二元向量取代，其中前面 14 位要滿足任兩個 1 之間至少有兩個 "0"，最多有 10 個 "0"；而加入的 3 位是用來確定整體上（兩個碼字間也要考慮）也滿足上述連續出現 "0" 的規定。最後，為了和諧，再加入 27 個 bits 來代表一個傳出的訊息。
13. 整體而言，一個代表 6 個 Ticks 所組成的訊息原有二元字母  $24 \times 8 = 192$  位，被編碼成  $33 \times 17 + 27 = 588$  位字母。

#### 解碼.

14.  $C_1$  可以更正 4 個 Erasures， $C_2$  可以更正一個錯誤及偵測出多個錯誤 (flagged)。
15. 由於 4-frame delay interleaving 的緣故，在  $C_1$  中所造成群聚長度 5 的錯誤，在整體而言含蓋了 17 行，於是所有的群聚長度在不大於  $15 \times 588 + 3 \times 17 = 8871$  的情況下都可以更正。也就是說影響的訊息 (Audio bits) 為  $15 \times 192 + 3 \times 8 = 2904$ ，在 CD 上，軌道長約為 0.25 公分。 $(n = 588)$
16. 6 個 Ticks (連續)  $L_1, R_1; L_2, R_2; \dots; L_6, R_6$  也可以再作安排以達成較好的解碼效果，例如， $L_1, L_3, L_5, R_1, R_3, R_5, Q_1, Q_2, L_2, L_4, L_6, R_2, R_4, R_6$ ，其中  $Q_1, Q_2$  為核對符號 (Bytes)。

## 10. Reed-Muller 及 Preparata 碼

在第三章中我們曾經介紹過特殊的 Reed-Muller 碼，在這裡將用兩種形式來介紹這一族出名的數碼。

$r$  階 Reed-Muller 碼 (遞迴建構) :  $RM(r, m)$  ,  $0 \leq r \leq m$  。

1.  $RM(0, m) = \{00 \cdots 0, 11 \cdots 1\}$  ,  $RM(m, m) = K^{2^m}$  ( $n = 2^m$ ) 。
2.  $RM(r, m) = \{(\mathbf{x}, \mathbf{x} + \mathbf{y}) \mid \mathbf{x} \in RM(r, m-1), \mathbf{y} \in RM(r-1, m-1)\}$  ,  $0 < r < m$  。

例.  $RM(0, 0) = \{0, 1\}$  ,  $RM(0, 1) = \{00, 11\}$  ,  $RM(0, 2) = \{0000, 1111\}$  ,  
 $RM(1, 1) = K^2$  ,  $RM(2, 2) = K^4$  ,  $RM(1, 2) = \{(\mathbf{x}, \mathbf{x} + \mathbf{y}) \mid \mathbf{x} \in K^2, \mathbf{y} \in \{00, 11\}\}$ 。  
 令  $G(r, m)$  為  $RM(r, m)$  的生成矩陣，則

$$G(r, m) = \begin{bmatrix} G(r, m-1) & G(r, m-1) \\ 0 & G(r-1, m-1) \end{bmatrix}, \text{ 其中}$$

$$G(0, m) = [11 \cdots 1]_{1 \times 2^m}$$

$$G(m, m) = \begin{bmatrix} G(m-1, m) \\ 00 \cdots 01 \end{bmatrix}$$

例.

$$G(1, 1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, G(1, 2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, G(1, 3) = \begin{bmatrix} G(1, 2) & G(1, 2) \\ 0 & G(0, 2) \end{bmatrix}。$$

**定理 10.1.** (Reed-Muller 碼的性質)

令  $RM(r, m)$  為  $(n, k, d)$  碼  $C$ 。則

- (1)  $n = 2^m$  。
- (2)  $d = 2^{m-r}$  。
- (3)  $k = \sum_{i=0}^r \binom{m}{i}$  。
- (4) 對於  $r > 0$  ,  $RM(r-1, m) \subseteq RM(r, m)$  。
- (5)  $C^\perp = RM(m-1-r, m)$  ,  $r < m$  。(  $C^\perp$  為  $C$  的對偶碼。 )

證明. ( 用歸納法 )

對於小的  $m$  可以直接驗證。

$$(4) \quad G(r-1, m) = \begin{bmatrix} G(r-1, m-1) & G(r-1, m-1) \\ 0 & G(r-2, m-1) \end{bmatrix}.$$

$$G(r, m) = \begin{bmatrix} G(r, m-1) & G(r, m-1) \\ 0 & G(r-1, m-1) \end{bmatrix}.$$

因為  $G(r-1, m-1)$  與  $G(r-2, m-1)$  分別為  $G(r, m-1)$  及  $G(r-1, m-1)$  的子矩陣 ( 歸納假設 ), 所以  $G(r-1, m)$  為  $G(r, m)$  的子矩陣, 故得證。 ■

(2) 因為  $RM(r-1, m-1) \subseteq RM(r, m-1)$ , 所以當  $\mathbf{x} \in RM(r, m-1)$  及  $\mathbf{y} \in RM(r-1, m-1)$  時,  $\mathbf{x} + \mathbf{y} \in RM(r, m-1)$ , 由歸納假設, 當  $\mathbf{x} \neq \mathbf{y}$  時,  $wt(\mathbf{x} + \mathbf{y}) \geq 2^{m-1-r}$ , 又  $wt(\mathbf{x}) \geq 2^{m-1-r}$ , 所以  $wt(\mathbf{x}, \mathbf{x} + \mathbf{y}) \geq 2^{m-r}$ 。另一方面, 如果  $\mathbf{x} = \mathbf{y}$ , 而  $\mathbf{y} \in RM(r-1, m-1)$ , 所以  $wt(\mathbf{y}) \geq 2^{(m-1)-(r-1)} = 2^{m-r}$ ,  $wt(\mathbf{y}, \bar{0}) \geq 2^{m-r}$ , 另外, 存在有一個  $\mathbf{y}$ , 它的加權為  $2^{m-r}$ , 所以  $d = 2^{m-r}$ 。 ■

(3) 從  $G(r, m)$  的定義,  $dimRM(r, m) = dimRM(r, m-1) + dimRM(r-1, m-1)$ 。於是, 由歸納假設推得  $dimRM(r, m) = \sum_{i=0}^r \binom{m}{i}$ 。 ■

(5) 由於  $dimRM(r, m) + dimRM(m-r-1, m) = 2^m$ , 所以只要證明  $RM(r, m)$  與  $RM(m-r-1, m)$  中的向量互相垂直即可。令  $(\mathbf{x}', \mathbf{x}' + \mathbf{y}')$  與  $(\mathbf{x}, \mathbf{x} + \mathbf{y})$  分別為這兩個數碼的向量, 所以

$$(\mathbf{x}', \mathbf{x}' + \mathbf{y}') \cdot (\mathbf{x}, \mathbf{x} + \mathbf{y}) = \mathbf{x}' \cdot \mathbf{x} + (\mathbf{x}' + \mathbf{y}') \cdot (\mathbf{x} + \mathbf{y}) = \mathbf{x}' \cdot \mathbf{y} + \mathbf{y}' \cdot \mathbf{x} + \mathbf{y}' \cdot \mathbf{y} = 0$$

( 歸納假設 ), (5) 得證。 ■

另一形式的  $RM(r, m)$  碼。

令  $K = GF(2)$ ，則  $K^m$  中的向量可以用以下的方式加以排列，一般稱它為標準排序 (Standard Ordering)。

$$\mathbf{a} = (a_0, a_1, a_2, \dots, a_{m-1}) \in K^m \Rightarrow o(\mathbf{a}) = \sum_{i=0}^{m-1} a_i 2^i \quad (*)$$

於是  $K^2$  的向量由小到大為  $(0, 0)$ ， $(1, 0)$ ， $(0, 1)$ ， $(1, 1)$ ，而  $K^3$  的向量排序為  $000, 100, 010, 110, 001, 101, 011, 111$ 。於是，任給一個由  $K^m$  對應到  $K$  的函數  $f$  都可以用有序  $2^m$  對來表示，也就是說用  $2^m$ -維向量  $\mathbf{v} = (f(00 \dots 00), f(10 \dots 00), \dots, f(11 \dots 11))$  來表示，或者是  $\mathbf{v} = (f(\mathbf{u}_0), f(\mathbf{u}_1), \dots, f(\mathbf{u}_{2^m-1}))$ ，其中  $o(\mathbf{u}_i) = i$ 。

為了建構  $RM(r, m)$  碼，我們先介紹一種特殊定義的函數。令  $1 \subseteq \{0, 1, 2, \dots, m-1\} = Z_m$ ，定義  $f_I(x_0, x_1, \dots, x_{m-1}) = \begin{cases} \prod_{i \in I} (x_i + 1), & I \neq \phi; \\ 1, & I = \phi. \end{cases}$

$f_I$  的向量表示法為  $\mathbf{v}_I$ 。

例.  $m = 3$ ， $I = \{1, 2\}$ ；則  $\mathbf{v}_I = (1, 1, 0, 0, 0, 0, 0, 0)$ 。

若  $I = \{0\}$  則  $\mathbf{v}_I = (1, 0, 1, 0, 1, 0, 1, 0)$ 。

**命題 10.2.**  $f_I(x_0, x_1, \dots, x_{m-1}) = 1$  若且唯若對於所有的  $i \in I$ ， $x_i = 0$ 。

**命題 10.3.** 令  $I, J$  為  $Z_m$  的兩個子集合，則  $\mathbf{v}_I \cdot \mathbf{v}_J = wt(v_{I \cup J}) \pmod{2}$ 。

**證明.** 首先，由定義，對於  $K^m$  中的任一個向量  $\mathbf{u}$ ， $f_I(\mathbf{u}_i) f_J(\mathbf{u}_i) = f_{I \cup J}(\mathbf{u}_i)$ 。  
(?) 因此， $\mathbf{v}_I \cdot \mathbf{v}_J = \sum_{i=0}^{2^m-1} f_I(\mathbf{u}_i) f_J(\mathbf{u}_i) = \sum_{i=0}^{2^m-1} f_{I \cup J}(\mathbf{u}_i) = wt(v_{I \cup J}) \pmod{2}$ 。現在，我們可以利用上述的觀念來定義一個  $RM(r, m)$  碼。 ■

**命題 10.4.** 令  $C$  為一個  $RM(r, m)$  碼，則  $C = \langle \{\mathbf{v}_I \mid I \subseteq Z_m \text{ 且 } |I| \leq r\} \rangle$ 。

**證明.** 先證明  $\{\mathbf{v}_I \mid I \subseteq Z_m \text{ 且 } |I| \leq r\}$  是一個線性獨立的集合，則  $C$  為維數  $k = \sum_{i=0}^r \binom{m}{i}$  的線性碼 (作業)。因此，利用以下的次序就可以求得  $G_{r,m}$ ：

當  $|I| < |J|$  或是在  $|I| = |J|$  的情況之下，存在一個  $j$  使得  $f_I(\mathbf{u}_j) < f_J(\mathbf{u}_j)$  且當  $i > j$  時  $f_I(\mathbf{u}_i) = f_J(\mathbf{u}_i)$ 。 ■

例.  $RM(3, 3)$

$$G_{3,3} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} \mathbf{v}_\phi \\ \mathbf{v}_{\{2\}} \\ \mathbf{v}_{\{1\}} \\ \mathbf{v}_{\{0\}} \\ \mathbf{v}_{\{1,2\}} \\ \mathbf{v}_{\{0,2\}} \\ \mathbf{v}_{\{0,1\}} \\ \mathbf{v}_{\{0,1,2\}} \end{matrix}$$

$$\binom{3}{0} + \binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 8$$

例.  $RM(4, 4)$

$$G_{4,4} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} \mathbf{v}_\phi \\ \mathbf{v}_{\{3\}} \\ \mathbf{v}_{\{2\}} \\ \mathbf{v}_{\{1\}} \\ \mathbf{v}_{\{0\}} \\ \vdots \\ \mathbf{v}_{\{0,1,2,3\}} \end{matrix}$$

$$\mathbf{v}_{\{0\}} = (f(0, 0, 0), f(1, 0, 0), f(0, 1, 0), f(1, 1, 0), f(0, 0, 1), f(1, 0, 1), f(0, 1, 1), f(1, 1, 1)) = (1, 0, 1, 0, 1, 0, 1, 0)$$

$$\mathbf{v}_{\{0,1\}} = (1, 0, 0, 0, 1, 0, 0, 0)$$

(註) 這個  $G_{3,3}$  和前面介紹的  $G(3, 3)$  一樣嗎？

### Preparata Codes.

用類似的概念我們可以建構出一種所謂的 Distance Invariant 的數碼。這個概念和強正則圖 (Strongly Regular Graphs) 有一些相似。

#### 定理 10.5. Distance Invariant (D.I.)

一個長度為  $n$  的數碼  $C$ ，如果  $C$  中的任兩個碼字  $\mathbf{v}$  與  $\mathbf{u}$  滿足對於  $1 \leq i \leq n$ ， $|\{\mathbf{w} | d(\mathbf{w}, \mathbf{v}) = i\}| = |\{\mathbf{z} | d(\mathbf{z}, \mathbf{u}) = i\}|$ ，則  $C$  為 D.I. 碼。

在建構我們所要的數碼之前，我們先介紹一個特殊函數  $x, x : GF(2^r) \rightarrow K$ 。

令  $U \subseteq GF(2^r)$ ，定義  $X(U) = (a_0, a_1, \dots, a_{2^r-1})$ ，其中

$$\begin{cases} a_i = 1 & , \text{ 如果 } \beta^i \in U, 0 \leq i \leq 2^r - 2; \\ a_{2^r-1} = 1 & , \text{ 如果 } 0 \in U; \quad \text{以及} \\ 0 & , \text{ 其它。} \end{cases}$$

例.

$$X(\{\beta^2, \beta^5, \beta^6\})(r=3) = (0, 0, 1, 0, 0, 1, 1, 0)。$$

$$X(\{0\}) = (0, 0, 0, 0, 0, 0, 0, 1)。$$

$$X(\phi) = (0, 0, 0, 0, 0, 0, 0, 0)。$$

#### 定義 10.6. (Preparata Codes) (Extended) $P(r)$

$P(r) = \{[X(U), X(V)] | U, V \in GF(2^r), |U| \text{ 和 } |V| \text{ 都是偶數}, \sum_{u \in U} u = \sum_{v \in V} v$

和  $\sum_{u \in U} u^3 + \left(\sum_{u \in U} u\right)^3 = \sum_{v \in V} v^3 \text{ 以及是奇數}\}$ 。

例.  $GF(2^3) = \mathbb{Z}_2[x]/\langle 1+x+x^3 \rangle$

$$U = \{\beta, \beta^2, \beta^5, 0\}, V = \{\beta^0, \beta, \beta^2, \beta^3, \beta^6, 0\}$$

$$\sum_{u \in U} u = \beta + \beta^2 + \beta^5 + 0 = 010 + 001 + 111 + 000 = 100 = \beta^0,$$

$$\sum_{v \in V} v = \beta^0 + \beta + \beta^2 + \beta^3 + \beta^6 + 0 = 100 + 010 + 001 + 110 + 101 + 000 = \beta^0,$$

$$\sum_{u \in U} u^3 = \beta^3 + \beta^6 + \beta + 0 = 110 + 101 + 010 + 000 = \beta^2,$$

$$\sum_{v \in V} v^3 = \beta^0 + \beta^3 + \beta^6 + \beta^2 + \beta^4 + 0 = 100 + 110 + 101 + 001 + 011 + 000 = \beta^6,$$

$$\text{因爲 } \beta^2 + (\beta^0)^3 = \beta^2 + 1 = \beta^6$$

$$\text{所以 } [X(U), X(V)] = [01100101, 11110011] \in P(3)。$$

(註)

(i)  $P(r)$  爲長度  $2^{r+1}$  的數碼。

(ii) "0" 是否爲  $U$  或  $V$  的元素，主要是用來造成  $|U|$  與  $|V|$  皆爲偶數，對於其它等式的驗證沒有影響。

爲了證明  $P(r)$  爲 D.I. 碼，我們需要以下的引理。

**引理 10.7.** 假設  $\alpha \in \sum_{u \in U} u$ ， $[X(U), X(V)] \in P(r)$  以及  $[X(A), X(B)] \in P(r)$ ，則  $[X(U \Delta A + \alpha), X(V \Delta B)] \in P(r)$ 。  
( $X \Delta Y = (X \setminus Y) \cup (Y \setminus X)$ ， $X + \alpha = \{x + \alpha \mid x \in X\}$ )

**證明.**

(1)  $|U \Delta A + \alpha|$  與  $|V \Delta B|$  爲偶數很容易驗證。

$$\begin{aligned} (2) \quad \sum_{x \in U \Delta A + \alpha} x &= \sum_{y \in U \Delta A} (y + \alpha) = \sum_{y \in U \Delta A} y + \alpha |U \Delta A| \\ &= \sum_{y \in U \Delta A} y = \sum_{y \in U} y + \sum_{y \in A} y = \sum_{y \in V} y + \sum_{y \in B} y = \sum_{y \in V \Delta B} y。 \end{aligned}$$

$$\begin{aligned} (3) \quad \sum_{x \in U \Delta A + \alpha} x^3 &+ \left( \sum_{x \in U \Delta A + \alpha} x \right)^3 \\ &= \sum_{y \in U \Delta A} (y + \alpha)^3 + \left( \sum_{y \in V \Delta B} y \right)^3 \\ &= \sum_{y \in U} (y + \alpha)^3 + \sum_{y \in A} (y + \alpha)^3 + \left( \sum_{y \in V} y + \sum_{y \in B} y \right)^3 \end{aligned}$$

$$\begin{aligned}
&= \sum_{y \in U} y^3 + \alpha \sum_{y \in U} y^2 + \alpha^2 \sum_{y \in U} y + \alpha^3 |U| + \sum_{y \in A} y^3 + \alpha \sum_{y \in A} y^2 + \alpha^2 \sum_{y \in A} y + \alpha^3 |A| \\
&\quad + \left( \sum_{y \in V} y \right)^3 + \left( \sum_{y \in V} y \right)^2 \left( \sum_{y \in B} y \right) + \left( \sum_{y \in V} y \right) \left( \sum_{y \in B} y \right)^2 + \left( \sum_{y \in B} y \right)^3 \\
&= \sum_{y \in V} y^3 + \sum_{y \in B} y^3 = \sum_{y \in V \Delta B} y^3. \quad \blacksquare
\end{aligned}$$

雖然 Extended Preparata 碼不是線性碼 (?)，但是它有一個和線性碼相類似的性質：如果在數碼中有  $\mathbf{0}$  這個碼字，則這個數碼的距離為不為  $\mathbf{0}$  碼字的最小加權，也就是說只要看碼字與  $\mathbf{0}$  的距離即可。

**推論 10.8.**  $\mathbf{0} \in P(r)$ ， $n \geq 3$ 。

**證明.** 因為  $[X(U), X(V)] \in P(r)$  可以推得  $[X(U \Delta U + \alpha), X(V \Delta V)]$  也是  $P(r)$  中的碼字，故得證。  $\blacksquare$

**推論 10.9.** 在  $P(r)$  中，如果  $d([X(U), X(V)], [X(A), X(B)]) = i$ ，則  $[X(U \Delta A + \alpha), X(V \Delta B)]$  的加權為  $i$ 。

**證明.** 因為  $d([X(U), X(V)], [X(A), X(B)]) = i$ ，所以  $d([X(U \Delta U + \alpha), X(V \Delta V)], [X(U \Delta A + \alpha), X(V \Delta B)]) = i$ 。

又因為  $[X(U \Delta U + \alpha), X(V \Delta V)] = \mathbf{0}$ ，故得證。  $\blacksquare$

由於距離  $i$  將反應到加權  $i$  的碼字數，這是一個定數，所以以下的性質就可以得到。

**命題 10.10.**  $P(r)$  是一個 D.I. 碼。

**定理 10.11.**  $d(P(r)) = 6$ ， $r \geq 3$ 。

**證明.** (摘要)

- (i)  $P(r)$  中有一個非零的碼字  $[X(U), X(V)]$ ，它的加權為  $d$ ， $d = wt(X(U)) + wt(X(V)) = |U| + |V|$  是偶數。
- (ii) 證明  $d \neq 2$ ， $d \neq 4$ 。(註)  $[X(U), X(V)] \in P(r) \Rightarrow [X(V), X(U)] \in P(r)$

(iii) 令  $U = \{0, u\}$  ,  $V = \{w, x, y, z\}$  它滿足

$w^3 = x^3 + y^3 + z^3$  ,  $u = w + x + y + z$  且  $u \neq 0$  。則  $d = 6$  。 ■

**定理 10.12.** 對於所有的奇數  $r \geq 3$  ,  $P(r)$  不是線性碼。

**證明.** 令  $U = \{0, u_1\}$  ,  $V = \{x_1, y_1, z_1, w_1\}$  ,  $A = \{0, u_2\}$  ,  $B = \{x_2, y_2, z_2, w_2\}$  , 則  $[X(A\Delta U + u_1), X(V\Delta B)]$  與  $[X(A\Delta U), X(V\Delta B)]$  的距離不大於 4 , 此為不可能 , 所以  $[X(A\Delta U), X(V\Delta B)] \in P(r)$  , 但是  $[X(A\Delta U), X(V\Delta B)] = [X(U), X(V)] + [X(A), X(B)]$  , 故得證。 ■

有關 Extended Preparata 碼的編碼與解碼方式在此省略。

## 11. 數碼與組合設計 (Codes and Designs)

從上一章建構  $P(r)$  的過程中，我們不難發現，利用集合可以適當地定義出數碼，以下的建構方式比起  $P(r)$  來說又更自然一些。

令  $S \subseteq Z_n$ ，則  $\mathbf{v}_S = (a_0, a_1, a_2, \dots, a_{n-1})$ ，其中當  $i \in S$  時， $a_i = 1$ ， $0 \leq i \leq n-1$ ，否則  $a_i = 0$ 。於是，任一個  $2^{Z_n}$  的子集合皆對應一個二元碼，它的長度為  $n$ 。

例.  $\{\{0, 1\}, \{0, 2\}, \{1, 2\}\} \subseteq 2^{Z_3}$ ，則對應的數碼  $C = \{(1, 1, 0), (1, 0, 1), (0, 1, 1)\} = \{110, 101, 011\}$ 。

**定義 11.1.** 設計 (Design)

一個定義於  $V = Z_n$  的設計為  $2^{Z_n}$  的一個子集合。

**定義 11.2.** 區組設計 (Block Design)

一個定義於  $V = Z_n$  的區組設計為  $\binom{Z_n}{k}$  的一個子集合，其中  $1 \leq k \leq n$ 。

**定義 11.3.** 不完全區組設計 (Incomplete Block Design)

$k < n$  的區組設計。

**定義 11.4.** 平衡不完全區組設計 (Balanced Incomplete Block Design)

在一個不完全設計中，如果任兩個元素出現在一樣多的區組中，則稱此設計為平衡不完全區組設計；出現的次數一般稱為頻率，以  $\lambda$  表示。

一個平衡不完全區組設計 (BIBD) 通常以  $(v, k, \lambda)$ -設計表示，其中  $|V| = v$ 。

**定義 11.5.** 配對平衡設計 (Pairwise Balanced Design, PBD)

如果只要平衡即可，區組並不一定要具有一樣多的元素個數，則稱它為配對平衡設計，以  $(v, k, \lambda)$ -設計表示， $K$  為所有可能區組大小所成的集合。

例.  $B = \{\{0, 1\}, \{0, 2\}, \{0, 3\}, \{0, 4\}, \{1, 2, 3, 4\}\}$  為一  $(5, \{2, 4\}, 1)$ -設計。

**定義 11.6.**  $t$ -設計 ( $t$ -design)

任意  $t$  個元素均出現在個  $\lambda$  區組中的設計，以  $t - (v, k, \lambda)$  設計表示。

所以 BIBD 為  $t = 2$  的設計。

**例.** 3-設計 ( $3 - (8, 4, 1)$  設計)

$\{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}, \{1, 2, 5, 6\}, \{1, 2, 7, 8\}, \{3, 4, 5, 6\}, \{3, 4, 7, 8\}, \{1, 3, 6, 8\},$   
 $\{1, 3, 5, 7\}, \{2, 4, 6, 8\}, \{2, 4, 5, 7\}, \{1, 4, 5, 8\}, \{1, 4, 6, 7\}, \{2, 3, 5, 8\}, \{2, 3, 6, 7\}\}。$

**命題 11.7.** 令  $B$  為一個  $t - (v, k, 1)$  設計，則  $B$  可以對應於一個長度為  $v$  距離不小於  $2(k - t + 1)$  的數碼。

**證明.** 任兩個區組最多只有  $t - 1$  個共同的元素。 ■

**說明：**由區組設計或  $t$ -設計所建構出來的二元碼每個碼字的加權都一樣，這種數碼又稱為是 "Code with Constant Weight"。在尋找  $A(n, d, w)$  時它們扮演非常重要的角色。

**定義 11.8.**  $A(n, d, w)$

$A(n, d, w) = \max \{ |C| \mid C \text{ 的長度為 } n, \text{ 距離 } d, \text{ 每個碼字的加權均為 } w \}。$

**例.**  $A(n, d, w) = 7$  ( 利用  $-STS(7) \approx (7, 3, 1)$  設計。 )

值得一提的是設計也可以由好的數碼建構出來 ( 隱藏在數碼中。 )

**定理 11.9.** 令  $C$  為一個完美的  $(n, k, d)$  碼，則在此數碼中加權為  $d$  的碼字所對應的集合形成一個  $(t + 1) - (n, d, 1)$  設計，其中  $t = \frac{d-1}{2}$ 。

**證明.** 由  $(t + 1)$ -設計的定義，我們只要證明任意  $t + 1$  個元素皆出現在唯一的  $d$ -集即可。

首先，因為  $C$  是一個完美碼，以碼字為中心，距離為  $t$  的球 (Ball) 可以把所有的  $n$ -維向量都含蓋在內，而且這些球是互斥的集合；於是，任意  $t + 1$  個元素所對應的向量  $w$  必然會落在某一 ( 唯一 ) 球內，這個球的中心 ( 碼字 )  $v$  它的加權也必定是  $d$  而且這個碼字所對應的集合也會包含所給定的那  $t + 1$  個元素。(?) 所以，定理得證。 ■

**例.** 哥雷碼中具  $G_{23}$  有一個  $4 - (23, 7, 1)$  設計。

其它相關的資料請參考 V.Pless 所著的 Introduction to the Theory of Error-Correcting Codes。

除了區組設計可以用來建構數碼外，我們也可以用垂直陣列來建構一個  $q$  元碼 ( $q$  不一定是 2)。

**定義 11.10.** 拉丁方陣 (Latin Square)

一個  $n \times n$  的陣列  $L = [l_{i,j}]$ ,  $l_{i,j} \in Z_n$ , 如果滿足  $Z_n$  中的每個元素在每一列都恰好出現一次, 則  $L$  為  $n$  階拉丁方陣。

**定義 11.11.** 垂直拉丁方陣 (Orthogonal Latin Squares)

兩個  $n$  階拉丁方陣  $L = [l_{i,j}]$ ,  $M = [m_{i,j}]$  互相垂直 (以  $L \perp M$  表示) 若且唯若  $\{(l_{i,j}, m_{i,j}) | 1 \leq i, j \leq n\} = Z_n^2$ 。

**定義 11.12.** 互相垂直的拉丁方陣 (Mutually Orthogonal Latin Squares, MOLS)

$L_1, L_2, \dots, L_t$  稱為是  $t$  個互相垂直的拉丁方陣若且唯若  $L_i \perp L_j$ ,  $1 \leq i \neq j \leq n$ 。

**定理 11.13.** 對於所有的質數次方數  $q$ , 階存在  $q-1$  個互相垂直的拉丁方陣。

**證明.** 令  $K = GF(q)$ ,  $K^* = GF(q) \setminus \{0\}$ , 則當  $s \in K^*$ , 令  $L^{(s)} = [l_{i,j}^{(s)}] = [is+j]$ ,  $L^{(1)}, L^{(2)}, \dots, L^{(q-1)}$  為所求。 ■

利用互相垂直的拉丁方陣可以製造一個垂直陣列 (Orthogonal Array)。

例.

|   |   |   |
|---|---|---|
| 0 | 1 | 2 |
| 2 | 0 | 1 |
| 1 | 2 | 0 |

 $\perp$ 

|   |   |   |
|---|---|---|
| 0 | 2 | 1 |
| 2 | 1 | 0 |
| 1 | 0 | 2 |

$\Rightarrow$

|           |   |   |   |   |   |   |   |   |   |
|-----------|---|---|---|---|---|---|---|---|---|
| 列         | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 2 |
| 行         | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 |
| $L^{(1)}$ | 0 | 1 | 2 | 2 | 0 | 1 | 1 | 2 | 0 |
| $L^{(2)}$ | 0 | 2 | 1 | 2 | 1 | 0 | 1 | 0 | 2 |

**定義 11.14.** 垂直陣列 (Orthogonal Array)

一個深度  $s$  的  $n$  維垂直陣列,  $(s, n)$ -垂直陣列, 是一個  $s \times n^2$  的陣列, 它滿足任兩列所造成的  $n^2$  個有序數 (上, 下) 對都相異。

**定理 11.15.** 一個  $(s, n)$ -垂直陣列存在的充要條件為  $s-2$  個互相垂直的  $n$  階拉丁方陣存在。

**定理 11.16.** 由一個  $(s, n)$ -垂直陣列存在可以建構一個長度  $s$  , 距離  $\delta - 1$  的  $n$  元碼。

**例.** 上例所造成的行向量為 3 元碼 , 長度 4 , 距離 3 , 一共有 9 個碼字。

**定義 11.17.** 部分垂直陣列 (Partial Orthogonal Array)

在定義 11.14 中 ,  $n^2$  改成  $m \leq n^2$  即可。

(註)

- (i) 部分垂直陣列也可以製造數碼。
- (ii) 垂直陣列有些學者翻譯成正交表。

## 12. 編碼理論指定作業

### Assignment 1

1. 對於任意的正整數  $m$  以及質數  $p$  , 試找出一個佈於  $Z_p$  而且不可分解的  $m$  次多項式。
2. 令  $A(n, d)$  代表長度是  $n$ , 距離為  $d$  的數碼中最多的碼字數 , 試求  $A(8, 3)$  ,  $A(9, 3)$  及  $A(10, 3)$  。

### Assignment 2

1. 在  $G_{24}$  中 , 若是收到  $\mathbf{w} = 0001\ 1100\ 0111\ 0110\ 1101\ 0000$  , 則正確的碼字是什麼 ?
2. 解微分方程式 :

$$A(z) \cdot (1 + nz) = -(1 - z^2) \cdot A'(z) + (1 + z)^n, A(0) = 1.$$

### Assignment 3

1. 利用 Decoding Algorithm for Linear Cyclic Codes 來解碼。

$$(n = 15, g(x) = 1 + x^4 + x^6 + x^7 + x^8).$$

Decode 001 000 000 110 000.

2. 利用 BCH 碼的解碼演算法來解碼。

$$(n = 15, g(x) = (1 + x + x^4)(1 + x^3 + x^4)).$$

Decode (1) 110 111 011 101 100 and (2) 110 011 100 100 000.

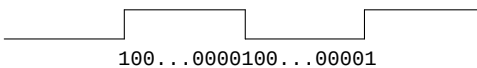
### Assignment 4

1. 令  $C$  為  $RS(2^4, 7)$  碼 ,  $g(x) = (1+x) \cdot (\beta+x) \cdot (\beta^2+x) \cdot (\beta^3+x) \cdot (\beta^4+x) \cdot (\beta^5+x)$ , 而  $GF(2^4)$  所用的多項式為  $1 + x + x^4$ 。

Decode (1)  $0\beta^3\beta\beta^5\beta^3\beta^2\beta^6\beta^{10}\beta 000\ 000$  and (2)  $\beta 0\beta^7 0\beta^{12}\beta^3\beta^3 10\ 000\ 000$ .

2. 利用 Transform 的概念解碼，問題 (1) 與 (2) 同上題。

### Assignment 5

1. 討論 "s-frame delay" 能更正的群聚長度。
2. 令右圖中  連續 0 的個數  $t$  最少為  $k$ ，最多為  $j$ ，當  $n > j > k$  時，求  $\max |\{\mathbf{v} | j \geq t \geq k\}|$  ( $\mathbf{v} \in K^n$ )。

### Assignment 6

1. 證明  $\{v_I | I \subseteq Z_m \text{ 且 } |I| \leq r \leq m\}$  為一線性獨立的集合。
2. 當  $r \geq 3$  時，證明  $d(P(r)) = 6$ 。

Assignments ( 參考解答 )

Assignment 1

1-4 林琲琪

1.1 對於任意的正整數  $m$  以及質數  $p$  , 試找出一個佈於  $Z_p$  而且不可分解的  $m$  次多項式。

Sol.

Before we solve the problem, we need to give some definitions and lemmas.

**Definition 1 :**

An irreducible polynomial over a field having a primitive element as a root is called a primitive polynomial.

**Definition 2 :**

Given an extension field  $E$  of  $\mathbf{F}$ , and  $\alpha \in E$ , where  $\alpha$  is algebraic over  $\mathbf{F}$ . If an irreducible polynomial  $p(x) \in \mathbf{F}(x)$  such that  $p(\alpha) = 0$  having 1 as the coefficient of the highest power of  $x$  appears,  $p(x)$  is called a monic polynomial of  $\alpha$ .

**Definition 3 :**

$m(x)$  is called the minimal polynomial of an element  $\alpha \in GF(p^r)$  if  $m(x)$  is the monic polynomial of  $\alpha$  of smallest degree with coefficients in  $GF(p^r)$  that has  $\alpha$  as a root.

**Definition 4 :**

Given any  $s$  so that  $0 \leq s \leq p^m - 1$  and let  $r$  be the smallest number with  $p^m s \equiv s \pmod{p^m - 1}$ , where  $p$  is some prime and  $m \in N$ . The cyclotomic coset containing  $s$ , denoted by  $C_s$ , consists of  $\{s, ps, p^2s, \dots, p^r s\}$  where each  $p^i s$  is reduced  $\pmod{p^m - 1}$ .

**Lemma 1 :**

In a field  $\mathbf{F}$  of characteristic  $p$ ,  $(x \pm y)^p = x^p \pm y^p$  for any  $x$  and  $y$  (variables or elements in  $\mathbf{F}$ ).

**Lemma 2 :**

Let  $m(x)$  be the minimal polynomial of an element  $\alpha$  in a finite field  $GF(p^r)$ . Then the following facts hold.

- (i)  $m(x)$  is irreducible.
- (ii) If  $\alpha$  is a root of a polynomial  $f(x)$  with coefficient in  $GF(p^r)$ , then  $m(x)$  divides  $f(x)$ .
- (iii)  $m(x)$  divides  $x^{p^r} - x$ .
- (iv) If  $m(x)$  is primitive, then its degree is  $r$ . In any case the degree of  $m(x)$  is less than or equal to  $r$ .

**Lemma 3 :**

Every finite field has a primitive element.

**Lemma 4 :**

$GF(p^s) \subseteq GF(p^r)$  if  $s$  divides  $r$  and an element  $\alpha \in GF(p^r)$  is in  $GF(p^s)$   $\alpha^{p^r} = \alpha$ .

**Lemma 5 :**

Let  $f(x)$  be a polynomial over  $GF(p)$  and let  $\alpha$  be a root of  $f(x)$  of order  $n$  in the multiplicative group of some field  $\mathbf{F}$ . Let  $r$  be the smallest integer so that  $p^{r+1} \equiv 1 \pmod{n}$ . Then  $\alpha, \alpha^p, \alpha^{p^2}, \dots, \alpha^{p^r}$  are all distinct roots of  $f(x)$ .

By Lemma 3, we have a primitive element  $\alpha \in GF(p^m)$ . Let  $m(x)$  be its minimal polynomial. Now we must prove that if  $u$  is the element of the cyclotomic coset containing 1, then  $m(x) = \prod_{i \in C_u} (x - \alpha^i)$  and  $m(x)$  is what we want for problem 1.

**Proof :**

Since  $\alpha$  is a primitive element in  $GF(p^m)$ .

Then its order is  $p^m - 1$ .

Let  $m_1(x) = \prod_{i \in C_1} (x - \alpha^i)$ .

In Lemma 5,  $n$  is  $p^m - 1$  and  $r$  is  $m - 1$ .

$\Rightarrow \alpha, \alpha^p, \alpha^{p^2}, \dots, \alpha^{p^r}$  are all distinct roots of  $m(x)$ .

$\Rightarrow m_1(x) | m(x)$ .

Now,

$$\begin{aligned} m_1(x) &= \prod_{i \in C_1} (x - \alpha^i) \\ &= x^{r+1} - (\alpha, \alpha^p, \alpha^{p^2}, \dots, \alpha^{p^r})x^r + \dots + (-1)^{r+1}(\alpha^{1+p+p^2+\dots+p^r}) \end{aligned}$$

By Lemma 1, for each coefficient  $\beta$  of  $m_1(x)$ ,  $u^\beta = u$ .

By Lemma 4, all coefficient of  $m_1(x)$  are in  $GF(p)$ .

Therefore,  $m_1(x) = m(x)$ .

By Lemma 2(i),  $m(x)$  is irreducible.

Since  $m(x)$  has primitive element  $\alpha$  as a root,

$m(x)$  is a primitive polynomial.

By Lemma 2(iv), the degree of  $m(x)$  is  $m - 1$ .

Therefore,  $m(x)$  is what we want. ■

### For example :

For  $GF(2^4)$ , i.e.  $p = 2$ ,  $m = 4$ ,  $C_1 = \{1, 2, 4, 8\}$ .

The minimal polynomial of the primitive element  $\alpha$  is

$$\prod_{i \in C_1} (x - \alpha^i) = (x - \alpha)(x - \alpha^2)(x - \alpha^4)(x - \alpha^8) = x^4 + x^3 + 1. \quad \blacksquare$$

**1.2** 令  $A(n, d)$  代表長度是  $n$ , 距離為  $d$  的數碼中最多的碼字數, 試求  $A(8, 3)$ ,  $A(9, 3)$  及  $A(10, 3)$ 。

**Sol.** (1)  $A(8, 3) = 20$

Before we resolve this problem, we need to give some definitions and lemmas as follows:

**Definition 1 :**

An  $(n, M, d)$  code consists of  $M(\geq 1)$  binary vectors (called codewords) of length  $n$  such that the distance of any two codewords is at least  $d$ .

**Definition 2 :**

Let  $C$  be an  $(n, M, d)$  code. The weight distribution of  $C$  with respect to a vector  $u$  is the  $(n+1)$ -tuple of integers  $(W_i(u), i = 0, 1, \dots, n)$ , where  $W_i(u)$  is the number of codewords  $v \in C$  such that  $d(u, v) = i$ . )

**Definition 3 :**

Let  $C$  be an  $(n, M, d)$  code. The distance distribution of  $C$  is the  $(n+1)$ -tuple of rational numbers  $(A_0, A_1, \dots, A_n)$  defined by  $A_i = \frac{1}{M} \sum_{u \in C} W_i(u)$ ,  $i = 0, 1, \dots, n$ .

**Lemma 1 :**

Let  $C$  be an  $(n, M, d)$  code with distance distribution  $(A_0, A_1, \dots, A_n)$ . Then the quantities  $(B_0, B_1, \dots, B_n)$  are nonnegative, where  $B_i = M^{-1} \sum_{k=0}^n A_k K_k(i)$ ,  $k = 0, 1, \dots, n$ . and  $K_k$  is a Krawtchouk polynomial, defined by

$$K_k(t) = \sum (-1)^j \binom{n-t}{k-j} \binom{t}{j}, \quad k = 0, 1, \dots, n.$$

To apply Lemma 1, let  $C$  be an optimal code of length  $n$  and minimum distance  $d$ . Then  $M = A(n, d) = 1 + A_d + A_{d+1} + \dots + A_n$ .

Suppose  $L^*(n, d)$  is the optimal solution to the following linear programming problem : choose real variables  $A_d, A_{d+1}, \dots, A_n$  so as to maximize

$$L = A_d + A_{d+1} + \dots + A_n.$$

Subject to the constraints

$$A_i \geq 0, \quad i = d, \dots, n.$$

$$B_k \geq 0, k = 0, \dots, n.$$

$$\text{where } B_k = M^{-1} \left( K_k(0) + \sum_{i=d}^n A_i K_k(i) \right).$$

$$\text{Then plainly } A(n, d) \leq 1 + L^*(n, d).$$

**Lemma 2 :**

Let  $C$  be an  $(n, M, d)$  code with  $M = A(n, d)$ , and suppose that  $M$  is odd.  
Then  $B_k \geq M^2 \binom{n}{k}$ ,  $k = 0, 1, \dots, n$ .

**Lemma 3 :**

If  $d$  is even, then  $A(n, d) = A(n-1, d-1)$ .

**Lemma 4 :**

$$A(n, d) \leq 2A(n-1, d).$$

Now, we want to prove  $A(8, 3) = 20$ .

By Lemma 3,  $A(8, 3) = A(9, 4)$ .

First, we want to prove  $A(9, 4) \geq 20$ .

Since Golay [1] found a  $(9, 20, 4)$ -code.

```

000  011  110
000  000  000
101  011  000
101  100  001
101  000  110
100  110  010
010  011  010
011  110  000
011  000  101
100  001  101
000  110  101
110  010  100
011  010  010
110  000  011
000  101  011
001  101  100
110  101  000
010  100  110
010  010  011
011  111  111

```

$A(9, 4) \geq 20$ .

Second, we want to prove  $A(9, 4) \leq 20$ .

As usual let  $C^-$  be an  $(8, M, 3)$  code with  $M = A(8, 3) = A(9, 4)$ ; and let  $C$  be the  $(9, M, 4)$  extended even weighted code, which have distance distribution  $(A_0, A_1, \dots, A_9)$  with  $A_0 = 1$  and  $A_1 = A_2 = A_3 = A_5 = A_7 = A_9 = 0$ .

Using the linear programming we have decided previously, now we need to maximize  $A_4 + A_6 + A_8$  subject to  $A_i \geq 0$ ,  $B_k \geq 0$ , and  $A_8 \leq 1$ .

$$\Rightarrow A_4 + A_6 + A_8 \leq 20\frac{1}{3}$$

$$\Rightarrow M \leq 21$$

Suppose  $M = 21$ .

Then, by Lemma 2, we can replace  $B_k \geq 0$  by  $B_k \geq \frac{1}{441} \binom{9}{k}$ .

Since  $M$  is odd,  $A_s \leq \frac{20}{21}$ .

Hence in this case, in spite of the extra inequality, all constant terms occurring in the inequalities are multiplied by  $\frac{20}{21}$ , so  $M \leq 1 + \frac{20}{21}(20\frac{1}{3}) < 21$ .

Hence  $A(9, 4) = A(8, 3) = 20$ . ■

**Sol.** (2)  $A(9, 3) = 40$

It's known that  $A(9, 4) = 20$ .

By Lemma 4, it follows that  $A(9, 3) = A(10, 4) \leq 40$ .

Next,  $(10, 4)$ -code can be represented as the union of ten affine squares in  $F_2^{10}$ , which can be obtained from each other by cyclic shifts.

One of the squares consists of the following four words:

(1010000001),

(1100101100),

(0001010111),

(0111111010).

A cyclic shift over five places transforms a square one into a parallel one.

Hence, the code can also be described as the union of five affine code, which can be obtained from each other by shifting cyclically.

It's a matter of straightforward verification that this code has minimum distance four.

Thus we have proved  $A(10, 4) = A(9, 3) = 40$ . ■

**Sol.** (3)  $72 \leq A(10, 3) \leq 79$

**Definition 4 :**

$A_n[n, d, w]$  code is an  $(n, d)$  code in which all words have weight  $w$ .

**Definition 5 :**

$A[n, d, w]$  denotes the maximum number of codewords of an  $[n, d, w]$  code.

**Lemma 5 :**

Let  $C$  be an  $(n, d)$  code with  $d$  even and with weight distribution  $W_i$ ,  $i = 0, 1, \dots, n$ . Furthermore let  $P$  and  $Q$  be upper bounds for  $A[n, d, \frac{1}{2}d + 1]$  and  $A[n - \frac{1}{2}d + 1, d, \frac{1}{2}d + 1]$  respectively. Then

$$W_{n-\frac{1}{2}d-1} + (P - Q)W_{n-\frac{1}{2}d+1} + P \sum_{i=n-\frac{1}{2}d+2}^n W_i \leq P.$$

**Lemma 6 :**

Let  $C$  be an  $(n, d)$  code with weight distribution  $W_i$ ,  $i = 0, 1, \dots, n$ . Then  $\sum_{i=0}^n (n - i)W_i \leq n \cdot A(n - 1, d)$ .

It is known that  $A(10, 4) = 40$ .

By Lemma 4, it follows that  $A(10, 3) = A(11, 4) \leq 80$ .

But we will prove that this bound cannot be achieved.

Let  $C$  be an  $(11, 4)$  code with 80 words and with distance distribution  $A_i$ ,  $i = 0, 1, \dots, n$ . It can be assumed that only even weights occur.

Then the constraints in the linear program we've described previously are

$$\begin{aligned} A_0 &= 1, A_4 \geq 0, A_6 \geq 0, A_8 \geq 0, A_{10} \geq 0, \\ 11A_0 + 3A_4 - A_6 - 5A_8 - 9A_{10} &\geq 0, \\ 55A_0 - A_4 - 5A_6 + 7A_8 + 35A_{10} &\geq 0, \\ 165A_0 - 11A_4 + 5A_6 + 5A_8 - 75A_{10} &\geq 0, \\ 330A_0 - 6A_4 + 10A_6 - 22A_8 + 90A_{10} &\geq 0, \\ 462A_0 + 14A_4 - 10A_6 + 14A_8 - 42A_{10} &\geq 0, \end{aligned}$$

and

$$M = A_0 + A_4 - A_6 + A_8 + A_{10}$$

should be maximized.

With the additional inequalities

$$\begin{aligned} A_8 + 4A_{10} &\leq 17 \text{ (by Lemma 5), and} \\ 11A_0 + 7A_4 + 5A_6 + 3A_8 + A_{10} &\leq 440 \text{ (by Lemma 6),} \end{aligned}$$

the following unique optimal solution is found

$$A_0 = 1, A_4 = 34, A_6 = 28, A_8 = 17, A_{10} = 0, \text{ and } M = 80.$$

The most important conclusion is that there are no words in  $C$  at distance ten from each other.

Now consider the code of length 12 consisting of all 80 words of  $C$ , each with one extra bit which is always zero together with all 80 complements. It can be checked easily that this is a  $(12, 4)$  code with 160 words.

But Best [2] has proved that such a code cannot exist. Hence  $A(11, 4) = A(10, 3) \leq 79$ .

Since Julin [3] found a  $(10, 72, 3)$ -code.

Thus  $A(10, 3) \geq 72$ .

Hence we've proved  $72 \leq A(10, 3) \leq 79$ . ■

## References

- [1] M.J.E. Golay, "*Binary Coding*," IEEE Trans. Inform. Theory, vol. IT-4, pp.23-28, Sept. 1954.
- [2] M.R. Best, "*Binary Codes with a minimum Distance of Four*," IEEE Trans. Inform. Theory, vol. IT-26, pp.738-743, Nov. 1980.
- [3] D. Julin, "*Two improved block codes*," IEEE Trans. Inform. Theory, vol. IT-11, pp.459, July 1965.

## Assignment 2

**2.1** Decode  $\mathbf{w} = (0001\ 1100\ 0111\ 0110\ 1101\ 0000)$ .

**Sol.** We use decoding algorithm, IMLD, to solve the problem.

a. Compute  $\mathbf{s} = \mathbf{w} \cdot H$ .

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ - & - & - & - & - & - & - & - & - & - & - & - \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} I_{12} \\ - \\ B \end{bmatrix}$$

$$\therefore \mathbf{s} = \mathbf{w} \cdot H = (101101101010).$$

- b. Check whether  $wt(\mathbf{s}) \leq 3$ .

$$wt(\mathbf{s}) = 7 \geq 3.$$

Thus we continue the next step.

- c. Check whether there exists  $i \in \{1, 2, \dots, 12\}$  such that  $wt(\mathbf{s} + \mathbf{b}_i) \leq 2$ , where  $\mathbf{b}_i$  is the  $i$ -th row of  $B$ .

$$\begin{aligned} wt(\mathbf{s} + \mathbf{b}_1) &= 8, & wt(\mathbf{s} + \mathbf{b}_2) &= 4, & wt(\mathbf{s} + \mathbf{b}_3) &= 8, \\ wt(\mathbf{s} + \mathbf{b}_4) &= 6, & wt(\mathbf{s} + \mathbf{b}_5) &= 6, & wt(\mathbf{s} + \mathbf{b}_6) &= 8, \\ wt(\mathbf{s} + \mathbf{b}_7) &= 4, & wt(\mathbf{s} + \mathbf{b}_8) &= 8, & wt(\mathbf{s} + \mathbf{b}_9) &= 8, \\ wt(\mathbf{s} + \mathbf{b}_{10}) &= 4, & wt(\mathbf{s} + \mathbf{b}_{11}) &= 6, & wt(\mathbf{s} + \mathbf{b}_{12}) &= 4. \end{aligned}$$

Since  $wt(\mathbf{s} + \mathbf{b}_i) > 2$  for  $\forall i \in \{1, 2, \dots, 12\}$  thus we continue the next step.

- d. Compute  $\mathbf{s} \cdot B$

$$\mathbf{s} \cdot B = (111001111101).$$

- e. Check whether  $wt(\mathbf{s} \cdot B) \leq 3$ .

$$wt(\mathbf{s} \cdot B) = 9 > 3.$$

Therefore we continue the next step.

- f. Check whether exists  $i \in \{1, 2, \dots, 12\}$  such that  $wt(\mathbf{s} \cdot B + \mathbf{b}_i) \leq 2$ , where  $\mathbf{b}_i$  is the  $i$ -th row of  $B$ . If yes,  $\mathbf{e} = | \mathbf{e}_i | \mathbf{s} \cdot B + \mathbf{b}_i |$

$$\begin{aligned} wt(\mathbf{s} \cdot B + \mathbf{b}_1) &= 6, & wt(\mathbf{s} \cdot B + \mathbf{b}_2) &= 8, & wt(\mathbf{s} \cdot B + \mathbf{b}_3) &= 6, \\ wt(\mathbf{s} \cdot B + \mathbf{b}_4) &= 2, & wt(\mathbf{s} \cdot B + \mathbf{b}_5) &= 4, & wt(\mathbf{s} \cdot B + \mathbf{b}_6) &= 6, \\ wt(\mathbf{s} \cdot B + \mathbf{b}_7) &= 6, & wt(\mathbf{s} \cdot B + \mathbf{b}_8) &= 4, & wt(\mathbf{s} \cdot B + \mathbf{b}_9) &= 6, \\ wt(\mathbf{s} \cdot B + \mathbf{b}_{10}) &= 4, & wt(\mathbf{s} \cdot B + \mathbf{b}_{11}) &= 6, & wt(\mathbf{s} \cdot B + \mathbf{b}_{12}) &= 4. \end{aligned}$$

$$\therefore \mathbf{e} = | \mathbf{e}_4 | \mathbf{s} \cdot B + \mathbf{b}_4 | = 0001 \ 0000 \ 0000 \ 0000 \ 0101 \ 0000$$

Thus the original message is  $\mathbf{w} + \mathbf{e}$ .

$$\mathbf{w} + \mathbf{e} = 0000 \ 1100 \ 0111 \ 0110 \ 1000 \ 0000. \quad \blacksquare$$

## 2.2 解微分方程：

$$A(z) \cdot (1 + nz) = -(1 - z^2) \cdot A'(z) + (1 + z)^n, A(0) = 1.$$

**Sol.**

**Lemma 1 :**

If  $\frac{dx}{dy} + a(x)y = f(x)$ , then  $y = [\int f(x)e^{\int a(x)dx} dx + c]e^{-\int a(x)dx}$ .

$$A(z) \cdot (1 + nz) = -(1 - z^2) \cdot A'(z) + (1 + z)^n$$

$$\Rightarrow \frac{dA(z)}{dz} + \frac{1+nz}{1-z^2} A(z) = \frac{(1+z)^n}{1-z^2}.$$

By Lemma 1,

$$\begin{aligned} A(z) &= \left[ \int \frac{(1+z)^n}{1-z^2} \cdot e^{\int \frac{1+nz}{1-z^2} dz} dz + c \right] \cdot e^{-\int \frac{1+nz}{1-z^2} dz} \\ &= \left[ \int \frac{(1+z)^n}{1-z^2} \cdot \frac{(1+z)^{\frac{1-n}{2}}}{(1-z)^{\frac{1+n}{2}}} dz + c \right] \cdot (1+z)^{-\frac{1-n}{2}} \cdot (1-z)^{\frac{1+n}{2}} \\ &= \left[ \int \frac{(1+z)^{\frac{n-1}{2}}}{(1-z)^{\frac{n+3}{2}}} dz + c \right] \cdot (1+z)^{-\frac{1-n}{2}} \cdot (1-z)^{\frac{1+n}{2}} \\ &= \left[ \frac{1}{n+1} \cdot \frac{(1+z)^{\frac{n+1}{2}}}{(1-z)^{\frac{n+1}{2}}} + c \right] \cdot (1+z)^{-\frac{1-n}{2}} \cdot (1-z)^{\frac{1+n}{2}} \end{aligned}$$

Since  $A(0) = 1$ .

$$\therefore c = \frac{n}{n+1}.$$

$$\text{Therefore } A(z) = \frac{1}{n+1}(1+z)^n + \frac{n}{n+1}(1+z)^{\frac{n-1}{2}} \cdot (1-z)^{\frac{n+1}{2}}. \quad \blacksquare$$

### Assignment 3

3.1  $n = 15$ ,  $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ . Decode 001 000 000 110 000.

**Sol.**

$$d = 5, \mathbf{w} = 001\ 000\ 000\ 110\ 000, w(x) = x^2 + x^9 + x^{10}.$$

By "Decoding Algorithm for Linear Cyclic Codes",

- a.  $s(x) \equiv w(x)(\text{mod } g(x)) = 1 + x^4 + x^7.$
- b.  $s'_1(x) \equiv x \cdot s(x)(\text{mod } g(x)) = 1 + x + x^4 + x^5 + x^6 + x^7.$   
 $s'_2(x) \equiv x^2 \cdot s(x)(\text{mod } g(x)) = 1 + x + x^2 + x^4 + x^5.$   
 $s'_3(x) \equiv x^3 \cdot s(x)(\text{mod } g(x)) = x + x^2 + x^3 + x^5 + x^6.$   
 $s'_4(x) \equiv x^4 \cdot s(x)(\text{mod } g(x)) = x^2 + x^3 + x^4 + x^6 + x^7.$   
 $s'_5(x) \equiv x^5 \cdot s(x)(\text{mod } g(x)) = 1 + x^3 + x^5 + x^6.$   
 $s'_6(x) \equiv x^6 \cdot s(x)(\text{mod } g(x)) = x + x^4 + x^6 + x^7.$   
 $s'_7(x) \equiv x^7 \cdot s(x)(\text{mod } g(x)) = 1 + x^2 + x^4 + x^5 + x^6.$   
 $s'_8(x) \equiv x^8 \cdot s(x)(\text{mod } g(x)) = x + x^3 + x^5 + x^6 + x^7.$   
 $s'_9(x) \equiv x^9 \cdot s(x)(\text{mod } g(x)) = 1 + x^2.$
- c.  $s_9(x) = (101000)$
- d. Since  $wt(\mathbf{s}_9) = 2 \leq \lfloor (d-1)/2 \rfloor = 2$ .

$$\text{Thus } e(x) \equiv x^{15-9} \cdot s'_9(x)(\text{mod } (x^{15} + 1)) = x^6 + x^8.$$

Therefore,

$$c(x) = e(x) + w(x) = x^2 + x^6 + x^8 + x^9 + x^{10} \leftrightarrow 001\ 000\ 101\ 110\ 000. \quad \blacksquare$$

**3.2** Let  $n = 15$ ,  $C(x) = \langle 1 + x^4 + x^6 + x^7 + x^8 \rangle$ .

Decode (1) 110 111 011 101 100 and (2) 110 011 100 100 000.

**Sol.** (1)

$$\mathbf{w} = 110\ 111\ 011\ 101\ 100.$$

$$w(x) = 1 + x + x^3 + x^4 + x^5 + x^7 + x^8 + x^9 + x^{11} + x^{12}.$$

$$g(x) = 1 + x^4 + x^6 + x^7 + x^8$$

$$= (1 + x + x^4) \cdot (1 + x + x^2 + x^3 + x^4)$$

$$\therefore C = \langle m_1(x) \cdot m_3(x) \rangle$$

$$H = \begin{bmatrix} 1 & 1 \\ \beta & \beta^3 \\ \beta^2 & \beta^6 \\ \beta^3 & \beta^9 \\ \beta^4 & \beta^{12} \\ \beta^5 & 1 \\ \beta^6 & \beta^3 \\ \beta^7 & \beta^6 \\ \beta^8 & \beta^9 \\ \beta^9 & \beta^{12} \\ \beta^{10} & 1 \\ \beta^{11} & \beta^3 \\ \beta^{12} & \beta^6 \\ \beta^{13} & \beta^9 \\ \beta^{14} & \beta^{12} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

By "Algorithm Decoding 2 Error-Correcting BCH Code (IMID)",

a.  $w(x) = 1 + x + x^3 + x^4 + x^5 + x^7 + x^8 + x^9 + x^{11} + x^{12}.$

$$s_1(x) \equiv w(\beta) = 1101.$$

$$s_3(x) \equiv w(\beta^3) = 0000.$$

b. Check whether  $s_1 = s_3 = \mathbf{0}$ .  $\rightarrow$  No, continue.

c. Check whether  $s_1 = \mathbf{0}$ ,  $s_3 \neq \mathbf{0}$ .  $\rightarrow$  No, continue.

d. Check whether  $s_1^3 = s_3$ .  $\rightarrow$  No, continue.

e. Check whether  $y^2 + s_1y + (s_3/s_1) + s_1^2 = 0$  has two different roots.

$$y^2 + \beta^7y + (0/\beta^7) + \beta^{14} = y^2 + \beta^7y + \beta^{14} = 0.$$

Let the above equation has two roots :  $\beta^i, \beta^j$ .

$$\Rightarrow \begin{cases} \beta^i + \beta^j = s_1(x) = \beta^7 \\ \beta^{i+j} = s_3(x)/s_1(x) + s_1^2(x) = \beta^{14} \end{cases}$$

$\Rightarrow$  The two roots are  $\beta^2$  and  $\beta^{12}$ .

$$\therefore e(x) = x^2 + x^{12}.$$

$$\text{Thus } c(x) = e(x) + w(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^7 + x^8 + x^9 + x^{11}$$

$$\leftrightarrow 111\ 111\ 011\ 101\ 000. \quad \blacksquare$$

**Sol.** (2)

By "Algorithm Decoding 2 Error-Correcting BCH Code (IMID)",

a.  $\mathbf{w} = 110\ 011\ 100\ 100\ 000$ .

$$w(x) = 1 + x + x^4 + x^5 + x^6 + x^9.$$

$$s_1(x) \equiv w(\beta) = 0000.$$

$$s_3(x) \equiv w(\beta^3) = 0000.$$

b. Check whether  $s_1 = s_3 = \mathbf{0}$ .  $\rightarrow$  Yes.

$$\text{Thus } \mathbf{c} = \mathbf{w} = 110\ 011\ 100\ 100\ 000. \quad \blacksquare$$

## Assignment 4

**4.1** For  $RS(2^4, 7)$  code,  $g(x) = (1+x) \cdot (\beta+1) \cdot (\beta^2+x) \cdot (\beta^3+x) \cdot (\beta^4+x) \cdot (\beta^5+x)$ ,  $m = -1$ ,  $t = 3$ ,  $GF(2^4) = \mathbb{Z}_2[x]/\langle 1+x+x^4 \rangle$ .

Decode (1)  $0\beta^3\beta\beta^5\beta^3\beta^2\beta^6\beta^{10}\beta 000\ 000$  and (2)  $\beta 0\beta^7 0\beta^{12}\beta^3\beta^3 10\ 000\ 000$ .

**Sol.** (1)

$$w(x) = \beta^3x + \beta x^2 + \beta^5x^3 + \beta^3x^4 + \beta^2x^5 + \beta^6x^6 + \beta^{10}x^7 + \beta x^8.$$

a.  $s_0(x) \equiv \mathbf{w}(\beta^0) = \beta^{14}.$

$$s_1(x) \equiv \mathbf{w}(\beta^1) = \beta^{12}.$$

$$s_2(x) \equiv \mathbf{w}(\beta^2) = \beta^{14}.$$

$$s_3(x) \equiv \mathbf{w}(\beta^3) = \beta^2.$$

$$s_4(x) \equiv \mathbf{w}(\beta^4) = \beta^0.$$

$$s_5(x) \equiv \mathbf{w}(\beta^5) = \beta^2.$$

b.  $M' = \left[ \begin{array}{ccc|c} \beta^{14} & \beta^{12} & \beta^{14} & \beta^2 \\ \beta^{12} & \beta^{14} & \beta^2 & \beta^0 \\ \beta^{14} & \beta^2 & \beta^0 & \beta^2 \end{array} \right]$

$$\sim \left[ \begin{array}{ccc|c} \beta^{14} & \beta^{12} & \beta^{14} & \beta^2 \\ 0 & \beta^{13} & \beta^9 & 0 \\ 0 & \beta^7 & \beta^3 & 0 \end{array} \right]$$

$$\sim \left[ \begin{array}{ccc|c} \beta^{14} & \beta^{12} & \beta^{14} & \beta^2 \\ 0 & \beta^{13} & \beta^9 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

c. Since  $\text{rank}(M') = 2$

$$\therefore \mathbf{e} = 2.$$

$$\begin{bmatrix} \beta^{14} & \beta^{12} \\ \beta^{12} & \beta^{14} \end{bmatrix} \begin{bmatrix} \sigma_0 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \beta^{14} \\ \beta^2 \end{bmatrix}$$

$$\Rightarrow \sigma_0 = \beta^7, \sigma_1 = \beta^{11}.$$

d.  $\sigma_A = \sigma_0 + \sigma_1 x + x^e = (a_1 + x)(a_2 + x) = \beta^7 + \beta^{11}x + x^2$   
 $\Rightarrow a_1 = \beta^1, a_2 = \beta^6 \quad \dots\dots\dots$  (Error Location).

e.  $\therefore \begin{bmatrix} 1 & 1 \\ \beta & \beta^6 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \beta^{14} \\ \beta^{12} \end{bmatrix}$

$\therefore b_1 = \beta^3, b_2 = \beta^0 \quad \dots\dots\dots$  (Error Location Number).

Thus  $\mathbf{c} = \mathbf{e} + \mathbf{w}$

$$= 0\beta^3\beta\beta^5\beta^3\beta^2\beta^6\beta^{10}\beta 000\ 000 + 0\beta^3 0\ 000\ \beta^0 00\ 000\ 000$$

$$= 00\beta\beta^5\beta^3\beta^2\beta^{13}\beta^{10}\beta 0000\ 000.$$

■

**Sol.** (2)

$$w(x) = \beta + \beta^7 x^2 + \beta^{12} x^4 + \beta^3 x^5 + \beta^3 x^6 + x^7.$$

a.  $s_0(x) \equiv \mathbf{w}(\beta^0) = \beta^{10}.$

$$s_1(x) \equiv \mathbf{w}(\beta^1) = \beta^{11}.$$

$$s_2(x) \equiv \mathbf{w}(\beta^2) = \beta^{12}.$$

$$s_3(x) \equiv \mathbf{w}(\beta^3) = \beta^{13}.$$

$$s_4(x) \equiv \mathbf{w}(\beta^4) = \beta^{14}.$$

$$s_5(x) \equiv \mathbf{w}(\beta^5) = \beta^0.$$

b.  $M' = \left[ \begin{array}{ccc|c} \beta^{10} & \beta^{11} & \beta^{12} & \beta^{13} \\ \beta^{11} & \beta^{12} & \beta^{13} & \beta^{14} \\ \beta^{12} & \beta^{13} & \beta^{14} & \beta^0 \end{array} \right]$

$$\sim \left[ \begin{array}{ccc|c} \beta^{10} & \beta^{11} & \beta^{12} & \beta^{13} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

c. Since  $rank(M') = 1$

$$\therefore \mathbf{e} = 1.$$

$$\therefore \beta^{10}\sigma_0 = \beta^{11}$$

$$\Rightarrow \sigma_0 = \beta.$$

d.  $\sigma_A = \sigma_0 + x^e = (a_1 + x) = \beta + x$

$$\Rightarrow a_1 = \beta^1 \quad \dots\dots (\text{Error Location}).$$

e. Since  $1 \cdot b_1 = \beta^{10}$ ,

$$\therefore b_1 = \beta^{10} \quad \dots\dots (\text{Error Location Number}).$$

Thus  $\mathbf{c} = \mathbf{e} + \mathbf{w}$

$$= \beta 0 \beta^7 0 \beta^{12} \beta^3 \beta^3 10 \text{ } 000 \text{ } 000 + 0 \beta^{10} 0 \text{ } 000 \text{ } 000 \text{ } 000 \text{ } 000$$

$$= \beta \beta^{10} \beta^7 0 \beta^{12} \beta^3 \beta^3 10 \text{ } 000 \text{ } 000.$$



**4.2** For  $RS(2^4, 7)$  code,  $g(x) = (1 + x) \cdot (\beta + 1) \cdot \dots \cdot (\beta^5 + x)$ ,  $m = -1$ ,  $t = 3$ ,  $GF(2^4) = Z_2[x]/\langle 1 + x + x^4 \rangle$ .

Decode (1)  $0\beta^3\beta\beta^5\beta^3\beta^2\beta^6\beta^{10}\beta 0000\ 000$  and (2)  $\beta 0\beta^7 0\beta^{12}\beta^3\beta^3 10\ 000\ 000$ .

**Sol.** (1)

$$w(x) = \beta^3 x + \beta x^2 + \beta^5 x^3 + \beta^3 x^4 + \beta^2 x^5 + \beta^6 x^6 + \beta^{10} x^7 + \beta x^8.$$

$$E_0(x) \equiv \mathbf{w}(\beta^0) = \beta^{14}.$$

$$E_{14}(x) \equiv \mathbf{w}(\beta^1) = \beta^{12}.$$

$$E_{13}(x) \equiv \mathbf{w}(\beta^2) = \beta^{14}.$$

$$E_{12}(x) \equiv \mathbf{w}(\beta^3) = \beta^2.$$

$$E_{11}(x) \equiv \mathbf{w}(\beta^4) = \beta^0.$$

$$E_{10}(x) \equiv \mathbf{w}(\beta^5) = \beta^2.$$

a. By 1.(1), we have  $\sigma_0 = \beta^7$ ,  $\sigma_1 = \beta^{11}$ .

b.  $E_9 = \beta^{11} E_{11} + \beta^7 E_{12} = \beta^8$ .

$$E_8 = \beta^{11} E_{10} + \beta^7 E_{11} = \beta^5.$$

$$E_7 = \beta^{11} E_9 + \beta^7 E_{10} = \beta^{14}.$$

$$E_6 = \beta^{11} E_8 + \beta^7 E_9 = \beta^4.$$

$$E_5 = \beta^{11} E_7 + \beta^7 E_8 = \beta^3.$$

$$E_4 = \beta^{11} E_6 + \beta^7 E_7 = \beta^{13}.$$

$$E_3 = \beta^{11} E_5 + \beta^7 E_6 = \beta^{10}.$$

$$E_2 = \beta^{11} E_4 + \beta^7 E_5 = \beta^{13}.$$

$$E_1 = \beta^{11} E_3 + \beta^7 E_4 = \beta^9.$$

$$c. E(x) = \sum_{k=0}^{14} E_k x^k$$

$$\Rightarrow (E_0, E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}, E_{12}, E_{13}, E_{14})$$

$$= (\beta^{14}, \beta^9, \beta^{13}, \beta^{10}, \beta^{13}, \beta^3, \beta^4, \beta^{14}, \beta^5, \beta^8, \beta^2, \beta^0, \beta^2, \beta^{14}, \beta^{12}).$$

d.  $\mathbf{e} = (E(\beta^0), E(\beta^1), \dots, E(\beta^{14}))$

$$= (0, \beta^3, 0, 0, 0, 0, \beta^0, 0, 0, 0, 0, 0, 0, 0, 0)$$

e.  $\mathbf{c} = \mathbf{w} + \mathbf{e}$

$$= (0, \beta^3, \beta, \beta^5, \beta^3, \beta^2, \beta^6, \beta^{10}, \beta, 0, 0, 0, 0, 0, 0)$$

$$+ (0, \beta^3, 0, 0, 0, 0, \beta^0, 0, 0, 0, 0, 0, 0, 0, 0)$$

$$= (0, 0, \beta, \beta^5, \beta^3, \beta^2, \beta^{13}, \beta^{10}, \beta, 0, 0, 0, 0, 0, 0).$$

■

**Sol.** (2)

$$w(x) = \beta + \beta^7 x^2 + \beta^{12} x^4 + \beta^3 x^5 + \beta^3 x^6 + x^7.$$

$$E_0(x) \equiv \mathbf{w}(\beta^0) = \beta^{10}.$$

$$E_{14}(x) \equiv \mathbf{w}(\beta^1) = \beta^{11}.$$

$$E_{13}(x) \equiv \mathbf{w}(\beta^2) = \beta^{12}.$$

$$E_{12}(x) \equiv \mathbf{w}(\beta^3) = \beta^{13}.$$

$$E_{11}(x) \equiv \mathbf{w}(\beta^4) = \beta^{14}.$$

$$E_{10}(x) \equiv \mathbf{w}(\beta^5) = \beta^0.$$

a. By 1.(2), we have  $\sigma_0 = \beta$ .

b.  $E_9 = \beta E_{11} = \beta^{14}$ .

$$E_8 = \beta E_{10} = \beta^0.$$

$$E_7 = \beta E_9 = \beta^1.$$

$$E_6 = \beta E_8 = \beta^0.$$

$$E_5 = \beta E_7 = \beta^1.$$

$$E_4 = \beta E_6 = \beta^2.$$

$$E_3 = \beta E_5 = \beta^1.$$

$$E_2 = \beta E_4 = \beta^2.$$

$$E_1 = \beta E_3 = \beta^3.$$

c.  $E(x) = \sum_{k=0}^{14} E_k x^k$

$$\Rightarrow (E_0, E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}, E_{12}, E_{13}, E_{14})$$

$$= (\beta^{10}, \beta^3, \beta^2, \beta^1, \beta^2, \beta^1, \beta^0, \beta^1, \beta^0, \beta^{14}, \beta^0, \beta^{14}, \beta^{13}, \beta^{12}, \beta^{11}).$$

d.  $\mathbf{e} = (E(\beta^0), E(\beta^1), \dots, E(\beta^{14}))$

$$= (0, \beta^{10}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

e.  $\mathbf{c} = \mathbf{w} + \mathbf{e}$

$$= (\beta, 0, \beta^7, 0, \beta^{12}, \beta^3, \beta^3, 1, 0, 0, 0, 0, 0, 0, 0)$$

$$+ (0, \beta^{10}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

$$= (\beta, \beta^{10}, \beta^7, 0, \beta^{12}, \beta^3, \beta^3, 1, 0, 0, 0, 0, 0, 0, 0).$$

■

## 5.1

**定理：** 假設每一個碼字最多只受到一個群聚型錯誤所影響，當  $C$  是可以更正  $l$  的群聚錯誤的數碼以及採 SFDI 型式傳遞，則可更正  $l(sn+1)$  的群聚型錯誤。

**證明.** 在此先以另一形式描述 SFDI

假設欲傳遞  $n$  個長度為  $n$  的碼字：

$$\begin{aligned} v_0 &= (v_{00}, v_{01}, \dots, v_{0(n-1)}) \\ v_1 &= (v_{10}, v_{11}, \dots, v_{1(n-1)}) \\ &\vdots \\ v_{(n-1)} &= (v_{(n-1)0}, v_{(n-1)1}, \dots, v_{(n-1)(n-1)}) \end{aligned}$$

先將  $v_{00}, v_{01}, \dots, v_{0(n-1)}, v_{10}, v_{11}, \dots, v_{(n-1)(n-1)}$  以下列方式 ” 放置 ” 在  $Z^2$  上： $v_{ij}$  放在  $(i + sj, j)$

傳遞訊息時以下列規則傳遞

1. 傳遞  $(0, 0)$  上的數字

傳遞  $(0, 1)$  上的數字

$\vdots$

傳遞  $(0, n-1)$  上的數字

傳遞  $(1, 0)$  上的數字

$\vdots$

傳遞  $(1, n-1)$  上的數字

$\vdots$

傳遞  $((n-1) + s(n-1), n-1)$  上的數字

2. 若點上無數字則可傳 ”\*” 或 ”0”。

由上述可知  $\{0 \leq x < n + (n-1)s, 0 \leq y < n\}$  中整數點  $(x, y)$  上數字被傳遞的次序為  $nx + y$

$\Rightarrow v_{ij}$  被傳遞的次序為  $n(i + sj) + j = ni + (ns + 1)j$

根據題目假設, SFDI 出錯

$\Rightarrow \exists i$  such that  $v_{i,x}, v_{i,x+1}, \dots, v_{i,x+l}$  皆傳錯

$\Rightarrow$  第  $ni + (ns + 1)x$  到第  $ni + (ns + 1)(x + l)$  個數字皆傳錯

$\Rightarrow$  出現大於等於  $(ns + 1)l + 1$  的群聚型錯誤

由反向推導可知, SFDI 可更正  $(ns + 1)l$  的群聚型錯誤。 ■

## 5.2 我們分兩種情況來回答這個問題。

- (1) 第一區間的 0 個數和最後一區間的 0 個數未受限制；舉例來說，101001000 的第一區間 0 個數為 0，最後一區間的 0 個數為 3。

以此定義來推導，可知  $00\cdots 0$  和  $100\cdots 0$ ， $010\cdots 0$ ， $001\cdots 0$ ， $\cdots$ ， $000\cdots 1$  都符合題目的要求且共  $n+1$  個，剩下的就是求 codeword 中有兩個以上的 1 的 codeword 數。

假設從第一個 1 到最後一個 1 一共有  $m$  個 bits，則對於固定的  $m$  而言，可再分成其中共有  $x$  個 1，for  $x = 2, 3, \cdots, m$ ；於是這個問題可以轉化成求 “ $b$  個數字總合為  $a$ ，但每個數字都必須介於  $c$  到  $d$  之間” 的組合數，令此組合數為  $B(a, b, c, d)$ ，則根據排容原理可寫出其公式如下：

$$B(a, b, c, d) = \sum_{i=0}^b (-1)^i \binom{b}{i} H_b^{a-bc-i(d-c+1)}, \text{ 其中 } H_y^x = C_x^{x+y-1}$$

$H_y^x$ ： “ $y$  個大於等於 0 的數字和為  $x$ ” 的組合數

$H_b^{a-bc}$ ： “ $b$  個大於等於  $c$  的數字和為  $a$ ” 的組合數

$H_b^{a-bc-i(d-c+1)}$ ： “ $b$  個大於等於  $c$  的數字和為  $a$ ，且其中至少有  $i$  個大於等於  $d+1$ ” 的組合數

因此題目所求的數目應為

$$1 + n + \sum_{m=2}^n \left( (n-m-1) \sum_{x=2}^m B(m-x, x-1, k, j) \right)$$

為了確定這個式子是對的，我寫了一個小程式來驗證，這個小程式一部份是計算上述公式的值，另一部份就是利用窮舉法找出所有符合要求的 codeword 並計算個數

```
combicount=0;
```

```
for (i = 0; i < n; i++) combilist[i]=0;
```

```
void combination (int level, int n, int a, int b)
```

```
{
```

```
    int i, prev, flag;
```

```

if (level < n) {
    combilist[level] = 0;
    combination (level+1, n, a, b);
    combilist[level] = 1;
    combination (level+1, n, a, b);
} else {
    for (flag=1, prev=-1, i = 0; i < n; i++) {
        if (combilist[i] == 1) {
            if (prev >= 0) {
                if (i - prev - 1 < a || i - prev - 1 > b) flag = 0;
            }
            prev = i;
        }
        if (flag) combicount++;
    }
}

```

呼叫此函數時使用  $\text{combination}(0, n, k, j)$  即可。

- (2) 第一區間的 0 個數和最後一區間的 0 個數也受限制。

對於有兩個以上 1 的 codeword，由於我們已有  $B$  函數可用，因此只要關心第一區間的  $a$  個 0 和最後一區間的  $b$  個 0 個數符合條件即可。

其數目為  $X_{\geq 2} = \sum_{\substack{j \geq a > k \\ j \geq b \geq k \\ n-a-b \geq 0}} \sum_{x=2}^{n-a-b} B(n-a-b-x, x-1, k, j)$  因為  $n > j$ ，

所以可確定  $00 \cdots 0$  不在考慮的範圍之內，因此只要再計算只有一個 1 的 codeword 數即可。

假設 codeword 中的  $n$  個 bits 的位置分別標明為  $1, 2, \dots, n$ ，若唯一的 1 在位置  $x$  上，則第一區間的 0 個數為  $x-1$  而最後一區間的 0 個數為  $n-x$ ，可求得以下關係

$$\begin{cases} j \leq x-1 \leq k \\ j \leq n-x \leq k \end{cases} \Rightarrow \begin{cases} j+1 \leq x \leq k+1 \\ n-k \leq x \leq n-j \end{cases}$$

因爲  $(n - k) - (j + 1) = (n - j) - (k + 1)$  , 所以可知當  $n - k \geq j + 1$  只須考慮  $n - k \leq x \leq k + 1$  即可 , 反之則考慮  $j + 1 \leq x \leq n - j$  即可

$$X_1 = \begin{cases} 2k - n + 2 & n - k \geq j + 1 \text{ 且 } 2k - n + 2 > 0 \\ n - 2j & n - k < j + 1 \text{ 且 } n - 2j > 0 \\ 0 & otherwise \end{cases}$$

爲了確定這個式子是對的，我又寫了一個小程式來驗證，這個小成是一部份是計算上述公式的值，另一部份就是利用窮舉法找出所有符合要求的 codeword 並計算個數

```

combicount=0;
for (i = 0; i < n; i++) combilist[i]=0;

void combination (int level, int n, int a, int b)
{
    int i, prev, flag;
    if (level < n) {
        combilist[level]=0;
        combination (level+1, n, a, b);
        combilist[level]=1;
        combination (level+1, n, a, b);
    } else {
        for (flag=1, prev=-1, i = 0; i < n; i++) {
            if (combilist[i]== 1) {
                if (prev >= 0) {
                    if (i-prev-1 < a || i-prev-1 > b) flag= 0;
                }
                prev= i;
            }
            if (n-prev-1 < a || n-prev-1 > b) flag= 0
            if (flag) combicount ++;
        }
    }
}

```

}

}

呼叫此函數時使用 `combination(0, n, k, j)` 即可。

## Assignment 6

6.1 證明  $\{v_I | I \subseteq Z_m \text{ 且 } |I| \leq r \leq m\}$  爲一線性獨立的集合。

**Proof.**

我們證明當  $r = m$  時成立。(若是  $r < m$ ，則所形成的集合爲一線性獨立集合的子集，自然也會是線性獨立的集合。)

首先，我們觀察向量  $v_I$  中底標最大的 "1"，令此底標爲  $w(v_I)$ ，例如， $w(v_\phi) = 2^m - 1$ ，而  $w(v_{Z_m}) = 0$ 。令  $I' = Z_m \setminus I = \{a_1, a_2, \dots, a_t\}$ ， $t \leq m$ ， $a_i \in Z_m$ 。我們證明  $w(v_I) = \sum_{i=1}^t 2^{a_i}$ 。這是因爲要使得  $f_I(u_j) = 1$ ， $u_j$  中底標爲  $I$  中元素的分量都要等於 0，而其它底標的分量可以任意選，所以最大底標出現在  $I'$  中的底標分量皆爲 1，於是  $w(v_I) = \sum_{i=1}^t 2^{a_i}$ 。

由於不同的集合  $I$  與  $J$  所得到的  $w(v_I) \neq w(v_J)$ ，所以只要把  $v_I$  依照  $w(v_I)$  值的大小由上往下排，則得到的矩陣形式如下：

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & \cdots & 1 & 1 \\ & & & & & 1 & \\ & A & & & 1 & & \\ & & & \ddots & & & \\ & & 1 & & & 0 & \\ & 1 & & & & & \\ 1 & 0 & 0 & 0 & \cdots & \cdots & 0 \end{bmatrix} \begin{matrix} v_\phi \\ \\ \\ \\ v_{Z_m} \end{matrix}$$

因此作業得証。 ■

**6.2** 當  $r \geq 3$  時，證明  $d(P(r)) = 6$ 。

答.

**引理 1.** 假設  $[X(U), X(V)]$  是  $P(r)$  的 codeword，則以下兩個也是  $P(r)$  的 codeword：

- (a)  $[X(V), X(U)]$
- (b)  $[X(U + \alpha), X(V + \alpha)]$ ，對任意  $\alpha \in GF(2^r)$ 。

**證明.**

- (a) (i)  $|V|$  和  $|U|$  都是偶數。

$$(ii) \sum_{v \in V} v = \sum_{u \in U} u。$$

$$(iii) \sum_{v \in V} v^3 + \left( \sum_{v \in V} v \right)^3 \quad (\text{因為 } [X(V), X(U)] \in P(r))$$

$$= \sum_{u \in U} u^3 + \left( \sum_{u \in U} u \right)^3 + \left( \sum_{v \in V} v \right)^3 \quad (\text{因為 } \sum_{v \in V} v = \sum_{u \in U} u)$$

$$= \sum_{u \in U} u^3$$

$$\Rightarrow [X(V), X(U)] \in P(r) \quad \blacksquare$$

- (b) (i)  $|V + \alpha|$  和  $|U + \alpha|$  都是偶數。

$$(ii) \sum_{u \in U + \alpha} u$$

$$= \sum_{u \in U} (u + \alpha)$$

$$= \sum_{u \in U} u + |U|\alpha \quad (\text{因為 } |U| \text{ 是偶數，所以 } |U|\alpha = 0)$$

$$= \sum_{u \in U} u = \sum_{v \in V} v$$

$$= \sum_{v \in V + \alpha} v$$

$$\begin{aligned}
& \text{(iii)} \quad \sum_{u \in U+\alpha} u^3 + \left( \sum_{u \in U+\alpha} u \right)^3 \\
&= \sum_{u \in U} (u + \alpha)^3 + \left( \sum_{u \in U} (u + \alpha) \right)^3 \\
&= \sum_{u \in U} (u + \alpha)^3 + \left( \sum_{u \in U} u \right)^3 \\
&= \sum_{u \in U} u^3 + \left( \sum_{u \in U} u \right)^3 + \alpha \sum_{u \in U} u^2 + \alpha^2 \sum_{u \in U} u + \alpha^3 |U| \\
&\quad \left( \text{因爲 } \sum_{u \in U} u^2 = \left( \sum_{u \in U} u \right)^2 = \sum_{v \in V} v^2 \right) \\
&= \sum_{v \in V} v^3 + \alpha \sum_{v \in V} v^2 + \alpha^2 \sum_{v \in V} v + \alpha^3 |V| \\
&= \sum_{v \in V} (v + \alpha)^3 = \sum_{v \in V+\alpha} v^3 \\
&\Rightarrow [X(V + \alpha), X(U + \alpha)] \in P(r) \quad \blacksquare
\end{aligned}$$

**推論 2.** 假設  $r$  為奇數，則對每一非零元素  $x \in GF(2^r)$ ，存在唯一的非零元素  $y$  使得  $y^3 = x$ 。

**證明.** 令  $\beta$  為  $GF(2^r)$  的 primitive element，假設存在  $\beta^a, \beta^b, \beta^c$  使得  $(\beta^a)^3 = (\beta^b)^3 = \beta^c$  且  $a \neq b$  則表示  $3a \equiv 3b \pmod{2^r - 1} \Rightarrow 3$  可整除  $2^r - 1$ 。但是因為  $2^r - 1 = 4 \times 2^{r-2} - 1 = (3+1) \times 2^{r-2} - 1$ ，所以可推知  $2^r - 1 \equiv 2^{r-2} - 1 \pmod{3} \Rightarrow 2^r - 1 \equiv 0 \pmod{3}$  for  $r \geq 2$  even,  $2^r - 1 \equiv 1 \pmod{3}$  for  $r \geq 1$  odd。可知前述存在  $\beta^a, \beta^b, \beta^c$  使得  $(\beta^a)^3 = (\beta^b)^3 = \beta^c$  且  $a \neq b$  和題目假設矛盾  $\Rightarrow$  證得唯一性。現在只要再證存在性即可。

令  $A = \{y^3 \mid y \in GF(2^r) \text{ 且 } y \neq 0\}$ ，因為  $y \neq 0 \Rightarrow y^3 \neq 0$  且根據前述唯一性證明，可知  $|A| = 2^r - 1$ 。

$\Rightarrow$  故得證。  $\blacksquare$

**定理.** 當  $r \geq 3$  時，同時  $r$  是 odd，證明  $d(P(r)) = 6$ 。

**證明.** 由於  $P(r)$  是 distance invariant，所以只要考慮非零 codeword 的 weight 即可。且因為已知  $P(r)$  中的 codeword 均為 even weight，所以只要證 weight 2 及 weight 4 的 codeword 不存在即可。

假設  $[X(V), X(U)] \in P(r)$  weight 為 2 : 因為  $|V|$  和  $|U|$  都是偶數 , 且根據引理 1 (a) , 假設  $|U| = 2, |V| = 0$  。再根據引理 1 (b) , 假設  $U = \{0, x\}$  且  $x \neq 0 \Rightarrow x + 0 \neq 0$  , 得到矛盾。

假設  $[X(V), X(U)] \in P(r)$  weight 為 4 : 因為  $|V|$  和  $|U|$  都是偶數 , 且根據引理 1 (a) , 假設  $|U| = 4, |V| = 0$  (  $|U| = 2, |V| = 2$  容後討論 ) 。根據引理 1 (b) , 假設  $U = \{0, x, y, z\}$  ,  $x, y, z$  非零且互不相等。則  $x^3 + y^3 + z^3 + (x + y + z)^3 = 0 \Rightarrow (x + y)(x + z)(y + z) = 0$  , 因為  $x, y, z$  互不相等故得到矛盾。

假設  $|U| = 2, |V| = 2$  。根據引理 1 (b) , 假設  $U = \{0, x\}, V = \{y, z\}$  ,  $y, z$  互不相等。則  $x^3 + x^3 = y^3 + z^3 \Rightarrow 0 = y^3 + z^3 \Rightarrow y^3 = z^3 \Rightarrow y = z$  ( 根據推論 2 ) 得到矛盾。

$\Rightarrow$  當  $r \geq 3$  時 , 證明  $d(P(r)) \geq 6$  。以下再證  $d(P(r)) = 6$  。

假設  $x, y \in GF(2^r)$  且  $x, y$  非零且互不相等 , 由推論 2 可得知存在唯一的非零  $z(x, y)$  使得  $z^3 = x^3 + y^3$  成立。如果對於所有符合假設的  $x, y$  都得不到  $x + y + z(x, y) \neq 0$  , 則表示  $x + y + z(x, y) = 0 \Rightarrow z(x, y) = x + y$  , 可得  $(x + y)^3 = x^3 + y^3 \Rightarrow x^2y + xy^2 = xy(x + y) = 0 \Rightarrow x + y = 0 \Rightarrow x = y$  , 由此矛盾可知必存在  $x, y$  非零且不相等 , 使得  $z^3 = x^3 + y^3$  及  $x + y + z \neq 0$  同時成立。此時若  $z = x$  , 則可推得  $y^3 = 0$  得到矛盾 , 故亦可知上述所得到的  $x, y, z$  非零且互不相等。解以下聯立方程式 for  $v, w$

$$\begin{cases} v + w = x + y + z \\ v^3 + w^3 = x^3 + y^3 + z^3 + (x + y + z)^3 \end{cases}$$

可知  $v$  可由  $(x + y + z)v^2 + (x + y + z)^2v + x^3 + y^3 + z^3 = 0$  解得 , 由於  $z^3 = x^3 + y^3$  及  $x + y + z \neq 0$  , 所以我們求得  $v = x + y + z \neq 0, w = 0$  為解  $\Rightarrow [X(\{0, x, y, z\}), X(\{0, x + y + z\})] \in P(r)$  , 故得證。 ■