

第三章 BCH码

本章内容

- 有限域
- BCH码的编码
- BCH码的译码
- 戈雷(Golay)码
- Reed-Solomon码

3.1 引言

- BCH码是一类最重要的循环码，能纠正多个随机错误，它是1959年由Bose、Chaudhuri及Hocquenghem各自独立发现的二元线性循环码，人们用他们的名字字头命名为BCH码。
- 在前面的讨论中，我们所做的只是构造一个码，然后计算它的最小距离，从而估计出它的纠错能力，而在BCH码中，我们将采用另外一种方法：先说明我们希望它能纠错的个数，然后构造这种码。

3.2 BCH码简述

- 若循环码的生成多项式具有如下形式:

$$g(x) = \text{LCM}[m_1(x), m_3(x), \dots, m_{2t-1}(x)]$$

其中LCM表示最小公倍式, t 为纠错个数, $m_i(x)$ 为素多项式, 则由此生成的循环码称为BCH码, 其最小码距 $d \geq d_0 = 2t + 1$ (d_0 称为设计码距), 它能纠正 t 个随机独立差错。

- BCH码的码长 $n = 2^m - 1$ 或是 $n = 2^m - 1$ 的因子

本原BCH码

非本原BCH码

- 例3.1: BCH(15,5)码, 可纠正3个随机独立差错, 即 $t=3$

$$d \geq d_0 = 2t+1 = 7$$

$$n=15=2^m-1, \text{ so } m=4$$

查不可约多项式表可得

$$m_1(x)=(23)_8=010011=x^4+x+1$$

$$m_3(x)=(37)_8=011111=x^4+x^3+x^2+x+1$$

$$m_5(x)=(07)_8=000111=x^2+x+1$$

$$\text{这样 } g(x)=\text{LCM}[m_1(x), m_3(x), m_5(x)]$$

$$=(x^4+x+1)(x^4+x^3+x^2+x+1)(x^2+x+1)$$

$$=x^{10}+x^8+x^5+x^4+x^2+x+1$$

➤ 例3.2: BCH(31,16)码, 可纠正3个随机独立差错, 即 $t=3$

$$d \geq d_0 = 2t + 1 = 7$$

$$n = 31 = 2^m - 1, \text{ so } m = 5$$

查不可约多项式表可得

$$m_1(x) = (45)_8 = 100101 = x^5 + x^2 + 1$$

$$m_3(x) = (75)_8 = 111101 = x^5 + x^4 + x^3 + x^2 + 1$$

$$m_5(x) = (67)_8 = 110111 = x^5 + x^4 + x^2 + x + 1$$

$$\text{这样 } g(x) = \text{LCM}[m_1(x), m_3(x), m_5(x)]$$

$$= x^{15} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^5 + x^3 + x^2 + x + 1$$

- 部分不可约多项式表

2阶	1	7				
3阶	1	13				
4阶	1	23	3	37	5	07
5阶	1	45	3	75	5	67

$n \leq 31$ 的本原BCH码

n	k	t	g(x)
7	4	1	13
15	11	1	23
15	7	2	721
15	5	3	2467
31	26	1	45
31	21	2	3551
31	16	3	107657
31	11	5	5423325
31	6	7	313365047

部分非本原BCH码

n	k	d	g(x)
17	9	5	727
21	16	3	43
21	12	5	1663
21	6	7	126357
21	4	9	643215
23	12	7	5343
25	5	5	4102041
27	9	3	1001001
27	7	6	7007007
33	6	7	3043

3.3 有限域

- 一个元素个数有限的域称为**有限域**，或者**伽罗华域**([Galois field](#));
- 有限域中元素的个数为一个素数，记为 $GF(p)$,其中 p 为素数;
- 一个大于1的整数，如果它的正因数只有1和它本身，就叫做**素数**，否则就叫做**合数**。
- 有限域中运算满足
 - 交换律: $a+b=b+a, a \cdot b=b \cdot a$
 - 结合律: $(a+b)+c=a+(b+c), a \cdot (b \cdot c)=(a \cdot b) \cdot c$
 - 和分配律: $a \cdot (b+c)=a \cdot b+a \cdot c$

- 可以将 $GF(p)$ 延伸为一个含有 p^m 个元素的域，称为 $GF(p)$ 的扩展域，表示为 $GF(p^m)$ ， m 是一个非零正整数。注意： $GF(p)$ 是 $GF(p^m)$ 的子集。
- 二进制域 $GF(2)$ 是扩展域 $GF(2^m)$ 的一个子域，类似于实数域是复数域的一个子域一样。除了数字0和1之外，在扩展域中还有特殊的元素，用一个新的符号 a 表示。 $GF(2^m)$ 中任何非0元素都可由 a 的幂次表示。
- 有限元素的集合 $GF(2^m)$ ，只能含有 2^m 个元素，并且对乘法封闭，其约束条件为： $a^{(2^m-1)} + 1 = 0$
- 根据这个多项式限制条件，任何幂次等于或超过 2^m-1 的域元素都可降阶为下述幂次小于 2^m-1 的元素：
$$a^{(2^m+n)} = a^{(2^m-1)} a^{n+1} = a^{n+1}$$
- 这样， $GF(2^m)$ 的元素可表示为：

$$GF(2^m) = \{0, a^0, a^1, a^2, \dots, a^{2^m-2}\}$$

扩展域GF(2^m)中的加法

- 在GF(2^m)中，将每个非0元素用多项式 $a_i(x)$ 表示，其系数至少有一个不为0。对于 $i=0,1,2,\dots,2^m-2$ ，有：

$$a^i = a_i(x) = a_{i,0} + a_{i,1}x + a_{i,2}x^2 + \dots + a_{i,m-1}x^{m-1}$$

- 考虑 $m=3$,有限域表示为GF(2³),下表为上式描述的基本元素 $\{x^0, x^1, x^2\}$ 映射为7个元素 $\{a^i\}$ 和一个0元素。表中的各行是二进制数字序列，代表上式中的系数 $a_{i,0}$ 、 $a_{i,1}$ 、 $a_{i,2}$ 的取值。

域元素	基本元素			
		x^0	x^1	x^2
	0	0	0	0
	a^0	1	0	0
	a^1	0	1	0
	a^2	0	0	1
	a^3	1	1	0
	a^4	0	1	1
	a^5	1	1	1
	a^6	1	0	1
	a^7	1	0	0

多项式为 $f(x)=1+x+x^3$ 的GF(8)的元素与基本元素之间的映射

- 有限域中两个元素的加法定义为两个多项式中同幂次项系数进行模2加，即

$$\mathbf{a^i+a^j=(a_{i,0}+a_{j,0})+(a_{i,1}+a_{j,1})x+\dots+(a_{i,m-1}+a_{j,m-1})x^{m-1}}$$

- 有限域的本原多项式**：因为这些函数用来定义有限域 $\text{GF}(2^m)$ 。

一个多项式是**本原多项式的充要条件**：一个 m 阶的不可约多项式 $f(x)$ ，如果 $f(x)$ 整除 x^n+1 的最小正整数 n 满足 $n=2^m-1$ ，则该多项式是本原的。

- 例3.3 本原多项式的辨别

(1) $p_1(x)=1+x+x^4$

(2) $p_2(x)=1+x+x^2+x^3+x^4$

分析: (1)通过验证这个幂次为 $m=4$ 的多项式是否能够整除 x^n+1 ,但不能整除 $1 \leq n < 15$ 范围内的 x^n+1 ，就可以确定它是否为本原多项式。经反复计算， $p_1(x)$ 是本原多项式， $p_2(x)$ 不是，因为它能整除 x^5+1 。

部分本原多项式

m		m	
3	$1 + x + x^3$	11	$1 + x^2 + x^{11}$
4	$1 + x + x^4$	12	$1 + x + x^4 + x^6 + x^{12}$
5	$1 + x^2 + x^5$	13	$1 + x + x^3 + x^4 + x^{13}$
6	$1 + x + x^6$	14	$1 + x + x^6 + x^{10} + x^{14}$
7	$1 + x^3 + x^7$	15	$1 + x + x^{15}$
8	$1 + x^2 + x^3 + x^4 + x^8$	16	$1 + x + x^3 + x^{12} + x^{16}$
9	$1 + x^4 + x^9$	17	$1 + x^3 + x^{17}$
10	$1 + x^3 + x^{10}$	18	$1 + x^7 + x^{18}$

考虑一个本原多项式定义的有限域的例子

- 选择 $p(x)=1+x+x^3$ ，多项式的幂次为 $m=3$ ，所以由 $p(x)$ 所定义的域中包含了 $2^m=2^3=8$ 个元素。求解 $p(x)$ 的根就是指找到 x 使 $p(x)=0$ 。我们所熟悉的二进制数0和1不能满足，因为 $p(1)=1, p(0)=1$ （运用模2运算）。由基本代数学理论我们知道，对于幂次为 m 的多项式必然有 m 个根。对于这个例子， $p(x)=0$ 有3个根，由于这3个根不可能位于与 $p(x)$ 系数相同的有限域中，而是位于扩展域 $GF(2^3)$ 中。用扩展域的元素 a 来定义多项式 $p(x)$ 的根，可写成如下形式： $p(a)=0$

即 $1+a+a^3=0 \Rightarrow a^3=1+a$

这意味着 a^3 可以表示为更低阶 a 项的加权和。

类似地有：

$$a^4=a*a^3=a*(1+a)=a+a^2$$

$$a^5=a*a^4=a*(a+a^2)=a^2+a^3=1+a+a^2$$

$$a^6=a*a^5=a*(1+a+a^2)=a+a^2+a^3=1+a^2$$

$$a^7=a*a^6=a*(1+a^2)=a+a^3=1=a^0$$

所以，有限域 $GF(2^3)$ 的8个元素为

$$\{0, a^0, a^1, a^2, a^3, a^4, a^5, a^6\}$$

- 这8个元素中哪些是 $p(x)=0$ 的3个根呢？我们可通过枚举找到！

$$p(a^0)=1, a^0 \text{不是}$$

$$p(a^1)=1+a+a^3=0, a^1 \text{是}$$

$$p(a^2)=1+a^2+a^6=1+a^0=0, a^2 \text{是}$$

$$p(a^3)=1+a^3+a^9=1+a^3+a^2=1+a^5=a^4, a^3 \text{不是}$$

$$p(a^4)=1+a^4+a^{12}=1+a^4+a^5=1+a^0=0, a^4 \text{是}$$

同理可计算 $p(a^5)$ 、 $p(a^6)$ 都不等于0，所以 $p(x)=1+x+x^3$ 的3个根是 a, a^2, a^4

$p(x)=1+x+x^3$, GF(8)加法运算表

+	a^0	a^1	a^2	a^3	a^4	a^5	a^6
a^0	0	a^3	a^6	a^1	a^5	a^4	a^2
a^1	a^3	0	a^4	a^0	a^2	a^6	a^5
a^2	a^6	a^4	0	a^5	a^1	a^3	a^0
a^3	a^1	a^0	a^5	0	a^6	a^2	a^4
a^4	a^5	a^2	a^1	a^6	0	a^0	a^3
a^5	a^4	a^6	a^3	a^2	a^0	0	a^1
a^6	a^2	a^5	a^0	a^4	a^3	a^1	0

$p(x)=1+x+x^3$, GF(8)乘法运算表

\times	a^0	a^1	a^2	a^3	a^4	a^5	a^6
a^0	a^0	a^1	a^2	a^3	a^4	a^5	a^6
a^1	a^1	a^2	a^3	a^4	a^5	a^6	a^0
a^2	a^2	a^3	a^4	a^5	a^6	a^0	a^1
a^3	a^3	a^4	a^5	a^6	a^0	a^1	a^2
a^4	a^4	a^5	a^6	a^0	a^1	a^2	a^3
a^5	a^5	a^6	a^0	a^1	a^2	a^3	a^4
a^6	a^6	a^0	a^1	a^2	a^3	a^4	a^5

- 如果 $GF(p)$ 上的所有元素(除0外)都可表示为某元素 a 的幂, 则 a 称为 $GF(p)$ 上的**本原元**。
- 例3.4 考虑 $GF(5)$, 因为 $p=5$ 是个素数, 模算数可以进行。考虑该域上的元素2,

$$2^0=1(\bmod 5)=1, 2^1=2(\bmod 5)=2$$

$$2^2=4(\bmod 5)=4, 2^3=8(\bmod 5)=3$$

因此, 所有 $GF(5)$ 上的非零元素, 即 $\{1,2,3,4\}$ 都可以表示成2的幂, 故2是 $GF(5)$ 上的本原元; 大家可以验证, 3也是 $GF(5)$ 上的本原元。

- $\text{GF}(p^m)$ 中，在模 $p(x)$ 运算下的扩域上， x 所表示的元素是本原元。
- 例如：用本原多项式 $p(x)=1+x+x^3$ 来构造 $\text{GF}(8)$ ，设 $\text{GF}(8)$ 上的本原元为 a ，通过将 a 的幂模 $p(a)$ 得到 $\text{GF}(8)$ 上的所有元素。

a 的幂	$\text{GF}(8)$ 上的元素
a^0	1
a^1	a
a^2	a^2
a^3	$a+1$
a^4	a^2+a
a^5	a^2+a+1
a^6	a^2+1

- 定理：设 b_1, b_2, \dots, b_{p-1} 为 $GF(p)$ 上的非零域元素，则 $x^{p-1}+1 = (x+b_1)(x+b_2)\dots(x+b_{p-1})$
- 从循环码知识我们知道，为了找到分组长度为 n 的循环码的生成多项式，首先分解 x^n+1 ，因此 x^n+1 可以表示为多个因子的乘积，即

$$x^n+1=f_1(x)f_2(x)\dots f_w(x)$$
- 在扩展域 $GF(p^m)$ 中， $n=p^m-1$

- 例3.5 考虑GF(2)和它的扩展域GF(8)。这里p=2,m=3,对 x^7+1 进行分解

$$x^7+1=(x+1)(x^3+x+1)(x^3+x^2+1)$$

同时我们知道，GF(8)中的非零元素为1, a , a+1, a², a²+1, a²+a, a²+a+1,因此我们可以写为

$$\begin{aligned}x^7+1 &= (x+1)(x+a)(x+a+1)(x+a^2)(x+a^2+1)(x+a^2+a)(x+a^2+a+1) \\ &= (x+1)[(x+a)(x+a^2)(x+a^2+a)][(x+a+1)(x+a^2+1)(x+a^2+a+1)]\end{aligned}$$

而在GF(8)上，有

$$x^3+x+1 = (x+a)(x+a^2)(x+a^2+a)$$

$$x^3+x^2+1 = (x+a+1)(x+a^2+1)(x+a^2+a+1)$$

极小多项式 $f_i(x)$	对应的根	元素用 a 的幂表示
$x+1$	1	a^0
x^3+x+1	a, a^2 和 a^2+a	a^1, a^2, a^4
x^3+x^2+1	$a+1, a^2+1$ 和 a^2+a+1	a^3, a^6, a^5

3.4 BCH码的编码

- 对一个分组长度 $n=p^m-1$ 、确定可纠 t 个错误的BCH码的生成多项式的步骤：

1. 选取一个次数为 m 的素多项式并构造 $GF(p^m)$
2. 求 $\alpha^i, i=0,1,2,\dots,n-2$ 的极小多项式 $f_i(x)$
3. 可纠 t 个错误的码的生成多项式为

$$g(x)=\text{LCM}[f_1(x),f_2(x),\dots,f_{2t}(x)]$$

用这种方法设计的码至少能纠 t 个错误，在很多情况下，这些码能纠多于 t 个错误！！因此 $d=2t+1$ 称为码的设计距离，其最小距离 $d^*\geq 2t+1$ 。注意：一旦确定了 n 和 t ，我们便可以确定BCH码的生成多项式。

- 例3.6 考虑GF(2)上的本原多项式 $p(a)=a^4+a+1$ ，我们将以此来构造GF(16),设a为本原元。GF(16)上以a的幂表示形式的元素及它们对应的极小多项式为：

a的幂	GF(16)的元素	极小多项式
a^0	1	$x+1$
a^1	a	x^4+x+1
a^2	a^2	x^4+x+1
a^3	a^3	$x^4+x^3+x^2+x+1$
a^4	$a+1$	x^4+x+1
a^5	a^2+a	x^2+x+1
a^6	a^3+a^2	$x^4+x^3+x^2+x+1$
a^7	a^3+a+1	x^4+x^3+1
a^8	a^2+1	x^4+x+1
a^9	a^3+a	$x^4+x^3+x^2+x+1$
a^{10}	a^2+a+1	x^2+x+1
a^{11}	a^3+a^2+a	x^4+x^3+1
a^{12}	a^3+a^2+a+1	$x^4+x^3+x^2+x+1$
a^{13}	a^3+a^2+1	x^4+x^3+1
2013/4/11 a^{14}	a^3+1	x^4+x^3+1

- 我们希望确定纠单错的BCH码的生成多项式，即 $t=1$ 且 $n=15$ 。由前面公式可知，一个BCH码的生成多项式由 $\text{LCM}[f_1(x), f_2(x), \dots, f_{2t}(x)]$ 给出，利用前面的表我们可获得最小多项式 $f_1(x)$ 和 $f_2(x)$ ，于是有：

$$\begin{aligned} g(x) &= \text{LCM}[f_1(x), f_2(x)] \\ &= \text{LCM}[(x^4+x+1), (x^4+x+1)] \\ &= x^4+x+1 \end{aligned}$$

因为 $\deg g(x)=n-k$ ，可得 $n-k=4$ ，所以 $k=11$ ，于是我们得到纠单一错误的BCH(15,11)码的生成多项式。该码的设计距离为 $d=2t+1=3$ ，可以计算该码的实际最小距离 d^* 也是3。

如果希望纠2个错误，且 $n=15$ 。则其生成多项式为

$$\begin{aligned}g(x) &= \text{LCM}[f_1(x), f_2(x), f_3(x), f_4(x)] \\&= \text{LCM}[(x^4+x+1), (x^4+x+1), (x^4+x^3+x^2+x+1), (x^4+x+1)] \\&= (x^4+x+1)(x^4+x^3+x^2+x+1) \\&= x^8+x^7+x^6+x^4+1\end{aligned}$$

因为 $\deg g(x)=n-k=8$ ，所以 $k=7$ ，于是我们得到纠2个错误的BCH(15,7)码的生成多项式。该码的设计距离为 $d=2t+1=5$ ，可以计算该码的实际最小距离 d^* 也是5。

如果希望纠3个错误，且 $n=15$ 。则其生成多项式为

$$g(x)=\text{LCM}[f_1(x),f_2(x),f_3(x),f_4(x),f_5(x),f_6(x)]$$

$$= (x^4+x+1)(x^4+x^3+x^2+x+1)(x^2+x+1)$$

$$= x^{10}+x^8+x^5+x^4+x^2+x+1$$

因为 $\deg g(x)=n-k=10$ ，所以 $k=5$ ，于是我们得到纠3个错误的BCH(15,5)码的生成多项式。该码的设计距离为 $d=2t+1=7$ ，可以计算该码的实际最小距离 d^* 也是7。

如果希望纠4个错误，且 $n=15$ 。则其生成多项式为

$$g(x)=\text{LCM}[f_1(x),f_2(x),f_3(x),f_4(x),f_5(x),f_6(x),f_7(x),f_8(x)]$$

$$= (x^4+x+1)(x^4+x^3+x^2+x+1)(x^2+x+1)(x^4+x^3+1)$$

$$= x^{14}+x^{13}+x^{12}+x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1$$

因为 $\deg g(x)=n-k=14$ ，所以 $k=1$ 。(简单的重复码)。于是我们得到纠4个错误的BCH(15,1)码的生成多项式。该码的设计距离为 $d=2t+1=9$ ，可以计算该码的实际最小距离 d^* 是15。在此情况下，设计距离不等于实际最小距离，码设计得太过度了，该码实际可纠 $(d^*-1)/2=7$ 个随机错误！

3.5 BCH码的译码

- 根据生成多项式，可以构造出快速的硬件编码器，而对于BCH码的译码，由于它是循环码的一个子类，任何对循环码的标准译码过程都适用于BCH码。下面我们主要讨论专门针对BCH码的更高效的算法：

Gorenstein-zierler译码算法

设 $c(x)$ 为发送码字多项式， $e(x)$ 为错误多项式，则接收到的多项式为 $r(x)=c(x)+e(x)$

设 y_1, y_2, \dots, y_w 为 $g(x)$ 在 $GF(p^m)$ 上的根，即 $g(y_i)=0, i=1, 2, \dots, w$ 。因为对某个信息多项式 $a(x)$ ，有 $c(x)=a(x)g(x)$ ，所以 $c(y_i)=0$

$$r(y_i)=c(y_i)+e(y_i)=e(y_i), i=1, 2, \dots, w$$

- 假设BCH码是根据一个域元素a来构造的, 考虑错误多项式

$$e(x) = e_{n-1}x^{n-1} + e_{n-2}x^{n-2} + \dots + e_1x + e_0$$

其中最多有t个系数为非零(可纠t个错误), 假设实际发生了v个错误, 其中 $0 \leq v \leq t$ 。设错误发生在位置 i_1, i_2, \dots, i_v , 则错误多项式可写为

$$e(x) = e_{i_1}x^{i_1} + e_{i_2}x^{i_2} + \dots + e_{i_v}x^{i_v}$$

其中 e_{i_k} 为第k个错误的大小, 对二元码, $e_{i_k} = 1$

对纠错问题，我们必须知道两件事：

(1)错误在哪里发生了，即错误的位置

(2)错误程度

因此，未知量为 i_1, i_2, \dots, i_v 和 $e_{i_1}, e_{i_2}, \dots, e_{i_v}$ ，它们分别表明错误发生的位置和程度。

伴随式可通过对接收到的关于 a 的多项式计算得到：

$$\begin{aligned} S_1 &= r(a) = c(a) + e(a) = e(a) \\ &= e_{i_1} a^{i_1} + e_{i_2} a^{i_2} + \dots + e_{i_v} a^{i_v} \end{aligned}$$

定义错误程度 $Y_k = e_{i_k}$ 和错误位置 $X_k = a^{i_k}$ ，

$k=1, 2, \dots, v$ 。其中 i_k 为第 k 个错误的位置， X_k 是与这个位置相关的域元素。

- 现在伴随多项式可写为

$$S_1 = Y_1 X_1 + Y_2 X_2 + \dots + Y_v X_v$$

对 $j=1, 2, \dots, 2t$, 我们定义伴随式

$$S_j = r(a^j) = c(a^j) + e(a^j) = e(a^j)$$

于是我们可得到 $2t$ 个联立方程组, 它有 v 个错误位置未知量 X_1, X_2, \dots, X_v 和 v 个错误程度未知量 Y_1, Y_2, \dots, Y_v :

$$\begin{cases} S_1 = Y_1 X_1 + Y_2 X_2 + \dots + Y_v X_v \\ S_2 = Y_1 X_1^2 + Y_2 X_2^2 + \dots + Y_v X_v^2 \\ \vdots \\ S_{2t} = Y_1 X_1^{2t} + Y_2 X_2^{2t} + \dots + Y_v X_v^{2t} \end{cases}$$

- 定义错误定位多项式

$$U(x) = U_v x^v + U_{v-1} x^{v-1} + \dots + U_1 x + 1$$

这个多项式的根是错误位置的逆 X_k^{-1} , $k=1,2,\dots,v$, 即

$$U(x) = (1 - xX_1)(1 - xX_2)\dots(1 - xX_v)$$

所以，如果我们知道错误定位多项式 $U(x)$ 的系数，便可以求得错误位置 X_1, X_2, \dots, X_v 。经过一系列代数变换，我们可得如下矩阵：

$$\begin{bmatrix} S_1 & S_2 & \cdots & S_{v-1} & S_v \\ S_2 & S_3 & \cdots & S_v & S_{v+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ S_v & S_{v+1} & \cdots & S_{2v-2} & S_{2v-1} \end{bmatrix} \begin{bmatrix} U_v \\ U_{v-1} \\ \vdots \\ U_1 \end{bmatrix} = \begin{bmatrix} S_{v+1} \\ S_{v+2} \\ \vdots \\ S_{2v} \end{bmatrix}$$

错误定位多项式的系数可通过对伴随式矩阵 M 求逆得到！

BCH码的译码步骤

1. 作为测试值，令 $v=t$ ，计算伴随矩阵 M 的行列式。如果行列式的值为零，令 $v=t-1$ ，再一次计算 M 的行列式。重复这个过程直到找到一个 v 值，使伴随矩阵的行列式不为0，该 v 值就是实际产生错误的数目。
2. 求 M 的逆，并计算错误定位多项式 $U(x)$ 的系数；
3. 求解 $U(x)=0$ 的零点，从中可计算错误位置 X_1, X_2, \dots, X_v 。如果是二元码，就到此为止(因为错误程度为1)；
4. 如果不是二元码，回到方程组解这些方程组就得到错误程度
$$\begin{cases} S_1 = Y_1X_1 + Y_2X_2 + \dots + Y_vX_v \\ S_2 = Y_1X_1^2 + Y_2X_2^2 + \dots + Y_vX_v^2 \\ \vdots \\ S_{2t} = Y_1X_1^{2t} + Y_2X_2^{2t} + \dots + Y_vX_v^{2t} \end{cases}$$

- 例3.7 考虑纠3个错误的BCH(15,5)码，它的生成多项式为

$$g(x)=x^{10}+x^8+x^5+x^4+x^2+x+1$$

设传输的是全0码字，接收到的多项式为 $r(x)=x^5+x^3$,故有两个错误分别在第4个位置和第6个位置，错误多项式为 $e(x)=x^5+x^3$ 。
 但译码器并不知道这些，它连实际发生了几个错误都不知道！

解：利用Gorenstein-aierler译码算法，首先用GF(16)上的算术计算出伴随式

$$S_1=a^5+a^3=a^{11}, \quad S_2=a^{10}+a^6=a^7$$

$$S_3=a^{15}+a^9=a^7, \quad S_4=a^{20}+a^{12}=a^{14}$$

$$S_5=a^{25}+a^{15}=a^5, \quad S_6=a^{30}+a^{18}=a^{14}$$

因为这是个纠3个错的码，首先令 $v=t=3$

$$M = \begin{bmatrix} S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \\ S_3 & S_4 & S_5 \end{bmatrix} = \begin{bmatrix} a^{11} & a^7 & a^7 \\ a^7 & a^7 & a^{14} \\ a^7 & a^{14} & a^5 \end{bmatrix}$$

Det(M)=0, 这表明发生的错误数少于3个。

下面令 $v=2$

$$M = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix} = \begin{bmatrix} a^{11} & a^7 \\ a^7 & a^7 \end{bmatrix}$$

Det(M) $\neq 0$, 这表明实际发生了2个错误。

下面计算 M^{-1}

$$M^{-1} = \begin{bmatrix} a^7 & a^7 \\ a^7 & a^{11} \end{bmatrix}$$

$$\begin{bmatrix} U_2 \\ U_1 \end{bmatrix} = M^{-1} \cdot \begin{bmatrix} S_3 \\ S_4 \end{bmatrix} = \begin{bmatrix} a^7 & a^7 \\ a^7 & a^{11} \end{bmatrix} \begin{bmatrix} a^7 \\ a^{14} \end{bmatrix}$$

求解 U_1 和 U_2 可得 $U_2=a^8$ 及 $U_1=a^{11}$, 从而

$$U(x)=a^8x^2+a^{11}x+1=(1+xa^5)(1+xa^3)$$

因此恢复出来的错误位置为 a^5 和 a^3 。因为该码是二元码，错误程度为1，故 $e(x)=x^5+x^3$ 。

#

3.6 戈雷(Golay)码

- 在第9页中，我们曾给出一些部分非本原BCH码的列表，Golay码就是(23,12)码。由表可查出，其生成多项式

$$(5343)_8 = 101\ 011\ 100\ 011$$

$$\text{即 } g_1(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1 \quad \longrightarrow$$

$$\text{或 } g_2(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1 \quad \longleftarrow$$

它们都是 $x^{23}+1$ 的因式，即 $x^{23}+1=(x+1)g_1(x)g_2(x)$

其最小码距为7，可纠正不大于3个的随机错误。

- Golay码是一个**完备码**。如果 r 位监督位所组成的校正子与误码图样一一对应，这种码组称为完备码。

- 定理：一个有M个码字，最小距离为 $2t+1$ 的 q -元 (n,k) 码，满足

$$M \left\{ \binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \dots + \binom{n}{t}(q-1)^t \right\} \leq q^n$$

其中 q^n 这个界称为汉明界，一个能到达汉明界的码称为完备码，即上式取等号。

- 容易证明：

$$2^{12} \left\{ \binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3} \right\} = 2^{23}$$

3.7 Reed-Solomon (RS)码

- 1960年MIT Lincoln实验室的S. Reed和G. Solomon在*Journal of the Society for Industrial and Applied Mathematics*上发表的一篇论文: **Polynomial Codes over Certain Finite Fields** (某些有限域上的多项式码)
- RS码的编码系统是建立在比特组基础上的, 即字节, 而不是单个的0和1, 因此它是非二进制BCH码, 这使得它处理突发错误特别好。

备注: 在许多现实生活的信道中, 错误不是随机的, 而是突发的。例如, 在一个移动通信信道中, 信号衰退导致突发错误。当错误连续发生时, 我们称它们为突发错误。

- 对于任意选取的正整数 s ，可构造一个相应码长为 $n=q^s-1$ 的 q 进制BCH码，其中码元符号取自有限域 $GF(q)$ ，而 q 为素数的幂。当 $s=1$ ， $q>2$ 时所建立的码长为 $n=q-1$ 的 q 进制BCH码，称为RS码。当 $q=2^m(m>1)$ ，码元符号取自域 $GF(2^m)$ 的二进制RS码可用来纠正突发错误。
- 输入信息分为 $k*m$ 比特一组，即每个符号有 m 比特， k 个符号形成一组。
- 一个可纠 t 个符号错误的RS码，有如下参数
 码长： $n=2^m-1$ 符号 或 $m(2^m-1)$ bit
 信息段： k 符号 或 km bit
 监督段： $n-k=2t$ 符号 或 $m(n-k)=2mt$ bit
 最小码距： $d=2t+1$ 符号 或 $md=m(2t+1)$ bit

例3.8 试构造一个能纠3个错误符号，码长 $n=15$, $m=4$ 的RS码。

解：已知 $t=3$, $n=15$, $m=4$, 所以有

码距： $d=2t+1=7$ 个符号(28bit) 监督段： $2t=6$ 个符号(24bit)

信息段： $n-6=9$ 个符号(36bit) 码长： $n=15$ 个符号(60bit)

因此该码是(15,9)RS码，也可看作是(60,36)二进制码；

最小距离为 d 的RS码生成多项式应具有如下形式：

$$g(x)=(x+a)(x+a^2)\dots(x+a^{d-1})$$

本例中， $d=7$

$$\begin{aligned} g(x) &= (x+a)(x+a^2)\dots(x+a^6) \\ &= x^6 + a^{10}x^5 + a^{14}x^4 + a^4x^3 + a^6x^2 + a^9x + a^6 \end{aligned}$$

其中 a^i 是GF(q)中的一个元素。

RS码生成多项式的次数总是 $2t$ ！