第十一章　群 与 编 码

必备知识：第九章。

在现代通信世界中，数据传送通常是端到端传输，它可以是从计算机终端与200英尺以外的大型机相互作用这种简单的工作(经由远离地球 20 000 英里高的轨道上的卫星传送信号实现)，或者是打一个电话或寄一封信到这个国家的另一个地区。数据传输的基本问题是收到送来的数据和收不到失真部分的数据。失真是由许多因素引起的。

编码理论为在数据传输中引入冗余信息来帮助检错、有时纠错而发展了各种技巧。其中某些技巧就是利用群论。

在数据传输过程中经常发生的另一类完全不同的问题，就是修改了发送出去的数据使得只有潜在的接收者才能重构原有数据。该问题可以追溯到古希腊早期。密码学是研究安全数据传输这种所谓密码系统技术的学科。随着电子商务和 ATM 的广泛应用，密码学已成为当今社会十分重要的学科。在前面一些章节中已给出了许多有关密码学的例子。在本章要对公钥密码学这一重要问题做一些简要介绍。

回　顾

美国数学家香农(Claude E. Shannon,1916－2001)在贝尔实验室工作，于1948年发表了一篇论文，描述了通信的数学理论，从此开创了信息论领域。此后不久，Richard Hamming 和他的同事在贝尔实验室奠定了纠错码的理论基础。

20 世纪上半叶，密码学的大部分研究工作主要由军方进行并服务于军队。1949 年，香农发表了论文“保密系统通信理论”，从此开创了密码学研究的新领域。该领域一直停滞不前，一直到1975年，美国斯坦福大学的两位研究人员发现了公钥密码学，这才导致密码学领域出现了大量的研究活动。1976年，美国麻省理工学院的三位研究人员发现了一个公钥密码系统，称为RSA (Rivest，Shamir 和 Adelman)系统并且得到了广泛的应用。今天，用得最广泛的系统是一个所谓的 DES 系统，它是一个对称密码系统。

11.1　二元信息码与检错码

信息的基本单位称作**报文**，它是一个有限字母表中字符的有限序列。可以选择字母表为集合 $B=\{0, 1\}$，于是要传送的每一个字符或符号可表示为 B 中的 m 个元素的序列，即：每个字

符或符号表示成二进制形式。信息的基本单位称作**字**，它是 m 个 0 和 1 的序列。

集合 B 在表 11.1 中给出的二元运算+下是一个群(见 9.4 节的例 5)。如果把 B 看成群 Z_2，那么+只是模 2 加法。从 9.5 节的定理 1 得到 $B^m=B\times B\times\cdots\times B$($m$ 个因子)在运算 $\oplus$ 下是一个群，其中 $\oplus$ 定义为

$$(x_1,x_2,\cdots,x_m)\oplus(y_1,y_2,\cdots,y_m)=(x_1+y_1,x_2+y_2,\cdots,x_m+y_m)$$

该群在 9.5 节的例 2 中已经讨论过，它的单位元是 $\overline{0}=(0,0,\cdots,0)$ 并且每个元素是它自身的逆。B^m 中的元素记为 $(b_1,b_2,\cdots,b_m)$ 或简记为 $b_1b_2\cdots b_m$。注意 B^m 有 2^m 个元素，即：群 B^m 的阶是 2^m。

图 11.1 给出了在传输通道上把一个字从某个点传输到另一点的基本过程。元素 $x\in B^m$ 通过传输通道传送并且作为元素 $x_t\in B^m$ 被接收。在实际操作中，由于天气干扰，电子问题等等使得传输通道也许会遭到干扰，通常称它为**噪声**，它可能引起 0 作为 1 接收或者完全相反。在传送一个字的过程中这种错误的数字传输可以上升到所收到的字不同于被发出的字，即 $x\neq x_t$。如果传输确实发生了错误，那么 x_t 可能是 B^m 的任意元素。

表 11.1

+	0	1
0	0	1
1	1	0

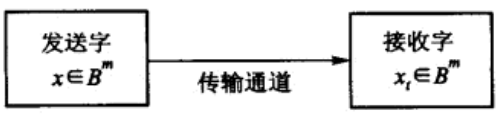

图 11.1

信息传输中的基本任务是减小收到的字不同于发送字的可能性。采用如下方法可做到这一点。首先选取一个整数 $n>m$ 和单射函数 $e:B^m\to B^n$。函数 e 称为一个(m,n)**编码函数**，并且可把它看做是 B^m 中的每个字表示成 B^n 中的一个字的一种方法。如果 $b\in B^m$，那么 $e(b)$称作表示 b 的**代码字**。附加的 0 和 1 能够提供一种方法检测或纠正在传输通道中所产生的错误。

现在通过传输通道来传送代码字，于是每个代码字 $x=e(b)$作为 B^n 中的字 x_t 被接收。这种情况如图 11.2 所示。

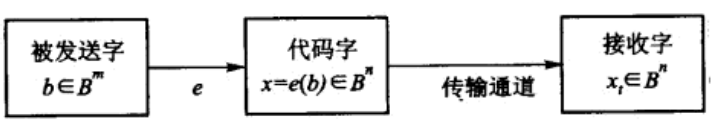

图 11.2

注意到要求编码函数 e 是单射，使得 B^m 中不同的字将被指定为不同的代码字。

如果传输通道是无噪声的，那么对 B^n 中的所有 x 有 $x_t=x$。在这种情况下，对每个 $b\in B^m$，$x=e(b)$都能接收到，因为 e 是一个已知函数，所以可以识别 b。

一般地，在传输中错误总是会发生的。如果 x 和 x_t 至少有一个但不超过 k 个位置不同，则称代码字 $x=e(b)$有 k 个或**更少的错误**传送。

设 $e: B^m \to B^n$ 是(m, n)编码函数，如果每当 $x=e(b)$有 k 个或更少的错误被传送时，那么 x_t 不是一个代码字(因此，x_t不可能是 x，所以不可能被正确地传输)，则说 e **检测到 k 个或更少的错误**。对于 $x \in B^n$，x 中 1 的个数称作 x 的**权**，并且用$|x|$表示。

例 1 在 B^5 中求下面每个字的权。

(a) x=01000 (b) x=11100 (c) x=00000 (d) x=11111

解 (a) $|x|=1$ (b) $|x|=3$ (c) $|x|=0$ (d) $|x|=5$ ■

例 2 (奇偶校验码) 下面的编码函数 $e: B^m \to B^{m+1}$ 称为奇偶$(m, m+1)$**校验码**：如果 $b=b_1b_2\cdots b_m \in B^m$，定义 $e(b)=b_1b_2\cdots b_mb_{m+1}$，其中

$$b_{m+1}=\begin{cases}0, & |b| \text{ 是偶数}\\ 1, & |b| \text{ 是奇数}\end{cases}$$

注意 b_{m+1} 为 0 当且仅当 b 中 1 的个数是一个偶数。于是得到每个代码字 $e(b)$有偶数权。在代码字的传输中仅一个错误将会使收到的字变成奇数权的字，所以它能被检测到。同理可以看到任何奇数个错误都能被检测。

为了具体地说明这种编码函数，设 m=3，于是

$$\left.\begin{array}{l}e(000)=0000\\ e(001)=0011\\ e(010)=0101\\ e(011)=0110\\ e(100)=1001\\ e(101)=1010\\ e(110)=1100\\ e(111)=1111\end{array}\right\}\text{代码字}$$

现在假设 b=111，那么 $x=e(b)$=1111。如果传输通道把 x 作为 x_t=1101 传送，那么$|x_t|$=3，于是可以知道出现了奇数个错误(至少一个)。 ■

应注意如果收到的字有偶数权，那么不能确定代码字是否被正确地传输，因为该编码函数不能检测偶数个错误。尽管受到这种限制，但奇偶校验码还是被广泛地采用。

例 3 考虑下面的$(m, 3m)$编码函数 e：$B^m \to B^{3m}$，如果

$$b=b_1b_2\cdots b_m \in B^m$$

定义

$$e(b)=e(b_1b_2\cdots b_m)=b_1b_2\cdots b_mb_1b_2\cdots b_mb_1b_2\cdots b_m$$

即：编码函数 e 把 B^m 中的每个字重复三次。对于一个具体的例子，设 m=3，那么

$$\left.\begin{array}{l} e(000)=000000000 \\ e(001)=001001001 \\ e(010)=010010010 \\ e(011)=011011011 \\ e(100)=100100100 \\ e(101)=101101101 \\ e(110)=110110110 \\ e(111)=111111111 \end{array}\right\}\text{代码字}$$

现在假设 b=011，那么 e(011)=011011011，还假设传输通道使下面加线的数字出错并收到字011111011。由于这不是代码字，所以检测到了错误。不难看到任意一个错误和两个错误都能够被检测到。■

设 x 和 y 是 B^m 中的字，在 x 与 y 之间的 **Hamming 距离** $\delta(x,y)$ 是指 $x\oplus y$ 的权 $|x\oplus y|$。因此，在 $x=x_1x_2\cdots x_m$ 与 $y=y_1y_2\cdots y_m$ 之间的距离是 $x_i\neq y_i$ 的 i 值的个数，即 x 与 y 不相同的位置数。因此，使用 $x\oplus y$ 的权计算不相同的位置数是一种很方便的方法。

例 4 求 x 与 y 之间的距离。

(a) x=110110, y=000101。

(b) x=001100, y=010110。

解 (a) $x\oplus y$=110011，所以 $|x\oplus y|=4$。

(b) $x\oplus y$=011010，所以 $|x\oplus y|=3$。■

定理 1(距离函数的性质) 设 x,y 和 z 是 B^m 中的元素，那么

(a) $\delta(x,y)=\delta(y,x)$

(b) $\delta(x,y)\geqslant 0$

(c) $\delta(x,y)=0$ 当且仅当 $x=y$

(d) $\delta(x,y)\leqslant\delta(x,z)+\delta(z,y)$

证明 性质(a)、(b)和(c)的证明是非常简单的，留给读者作为练习。

(d) 对于 B^m 中的 a 和 b，因为在 a 与 b 的任意位置上相异一定含有一个 1，所以

$$|a\oplus b|\leqslant|a|+|b|$$

此外，如果 $a\in B^m$，那么 $a\oplus a=\bar{0}$，即 B^m 中的单位元。于是

$$\begin{aligned}\delta(x,y)&=|x\oplus y|=|x\oplus\bar{0}\oplus y|=|x\oplus z\oplus z\oplus y|\\&\leqslant|x\oplus z|+|z\oplus y|=\delta(x,z)+\delta(z,y)\end{aligned}$$

■

编码函数 $e:B^m\to B^n$ 的**最短距离**是指所有不同对代码字之间距离的最小值，即

$$\min\left\{\delta(e(x),e(y))\middle|x,y\in B^m\right\}$$

例 5 考虑下面的(2,5)编码函数 e：

$$\left.\begin{aligned} e(00)&=00000\\ e(10)&=00111\\ e(01)&=01110\\ e(11)&=11111 \end{aligned}\right\}\text{代码字}$$

它的最短距离是 2，这能通过计算 6 对不同代码字之间距离的最小值得到验证。 ■

定理 2 一个(m, n)编码函数$e: B^m \to B^n$能够检测k个或更少错误当且仅当它的最短距离至少是$k+1$。

证明 假设在任意两个代码字之间的最短距离至少是 $k+1$。设 $b \in B^m$，$x=e(b) \in B^n$ 表示 b 的代码字，那么传送 x 并且收到的是 x_t。如果 x_t 是与 x 不同的代码字，那么 $\delta(x, x_t) \geqslant k+1$，所以传送 x 有 $k+1$ 个或更多的错误。因此，如果传送 x 只有 k 个或更少的错误，那么 x_t 不可能是一个代码字。这就推出 e 能检测 k 个或更少的错误。

反之，假设代码字之间的最短距离是 $r \leqslant k$，并且设 x 和 y 是有 $\delta(x, y)=r$ 的代码字。如果 $x_t=y$，即：如果传送 x 并且接收错误的 y，那么就会出现 $r \leqslant k$ 个错误并且它们没有被检测到。因此，e 能检测 k 个或更少错误是不成立的。 ■

例 6 考虑$(3, 8)$编码函数$e: B^3 \to B^8$定义为

$$\left.\begin{aligned} e(000)&=00000000\\ e(001)&=10111000\\ e(010)&=00101101\\ e(011)&=10010101\\ e(100)&=10100100\\ e(101)&=10001001\\ e(110)&=00011100\\ e(111)&=00110001 \end{aligned}\right\}\text{代码字}$$

问 e 能检测多少个错误？

解 e 的最短距离是 3，这能通过在所有 28 对不同代码字之间计算距离的最小值得到检验。由定理 2，代码能检测 k 个或更少错误当且仅当它的最短距离至少是 $k+1$。因为最短距离是 3，所以 $3 \geqslant k+1$，即 $k \leqslant 2$。因此，代码能检测两个或更少的错误。 ■

群码

到目前为止，还没有利用$(B^n, \oplus)$是一个群这一事实，下面将考虑利用 B^n 的这个性质的编码函数。

一个(m, n)编码函数$e: B^m \to B^n$称为一个**群码**，如果

$$e(B^m)=\{e(b) \mid b \in B^m\}=\mathrm{Ran}(e)$$

是 B^n 的一个子群。

回顾 9.4 节子群的定义，如果 N 满足下述条件，则 N 是 B^n 的一个子群：(a) B^n 的单位元在 N 中，(b)如果 x 和 y 属于 N，那么 $x \oplus y \in N$，(c) 如果 x 属于 N，那么它的逆属于 N。性质(c)不必在此检验，因为 B^n 中的每个元素都是它自身的逆。此外，因为 B^n 是阿贝尔群，所以 B^n 的每个子群是一个正规子群。

例 7 考虑(3, 6)编码函数 $e: B^3 \to B^6$ 定义为

$$\left.\begin{array}{l} e(000)=000000 \\ e(001)=001100 \\ e(010)=010011 \\ e(011)=011111 \\ e(100)=100101 \\ e(101)=101001 \\ e(110)=110110 \\ e(111)=111010 \end{array}\right\} \text{代码字}$$

表明该编码函数是一个群码。

解 必须表明所有代码字的集合

$$N=\{000000,\ 001100,\ 010011,\ 011111,\ 100101,\ 101001,\ 110110,\ 111010\}$$

是 B^6 的一个子群。首先注意到 B^6 的单位元属于 N，其次通过验证所有的可能性可以验证如果 x 和 y 是 N 中的元素，那么 $x \oplus y$ 是 N 中的元素。因此，N 是 B^6 的一个子群，且所给编码函数是一个群码。 ■

下面定理的证明方法同证明两个集合 A 和 B 相等，即证明 $A \subseteq B$ 且 $B \subseteq A$ 是相似的。这里通过证明 $\delta \leqslant \eta$ 和 $\eta \leqslant \delta$ 来证明 $\delta = \eta$。

定理 3 设 $e: B^m \to B^n$ 是一个群码，那么 e 的最短距离是非零代码字的最小权。

证明 设 δ 是群码的最短距离，并且假设 $\delta = \delta(x, y)$，其中 x 和 y 是不同的代码字。同样，设 η 是非零代码字的最小权，并且假设对一个代码字 z，$\eta = |z|$。因为 e 是一个群码，所以 $x \oplus y$ 是一个非零代码字。因此

$$\delta = \delta(x, y) = |x \oplus y| \geqslant \eta$$

另一方面，因为 0 和 z 是不同的代码字，所以

$$\eta = |z| = |z \oplus 0| = \delta(z, 0) \geqslant \delta$$

因此，$\eta = \delta$。 ■

下面的例子给出了群码的一个优点。

例 8 例 7 中的群码的最短距离是 2，因为由定理 3 可知该距离等于 7 个非零代码字中 1 的最小个数。直接检验将需要 28 种不同的计算。 ■

下面简要地看一看产生群码的过程。首先，需要几个关于布尔矩阵的附加结果。考虑表 11.1

中定义的具有+运算的集合 B。现在设 $\boldsymbol{D}=[d_{ij}]$，$\boldsymbol{E}=[e_{ij}]$都是 $m\times n$ 布尔矩阵，把**模 2 的和** $\boldsymbol{D}\oplus\boldsymbol{E}$ 定义为 $m\times n$ 布尔矩阵 $F=[f_{ij}]$，其中

$$f_{ij}=d_{ij}+e_{ij},\quad 1\leqslant i\leqslant m,\quad 1\leqslant j\leqslant n \text{ (这里的+是 } B \text{ 中的加法)}$$

例 9

$$\begin{bmatrix}1&0&1&1\\0&1&1&0\\1&0&0&1\end{bmatrix}\oplus\begin{bmatrix}1&1&0&1\\1&1&0&1\\0&1&1&1\end{bmatrix}=\begin{bmatrix}1+1&0+1&1+0&1+1\\0+1&1+1&1+0&0+1\\1+0&0+1&0+1&1+1\end{bmatrix}=\begin{bmatrix}0&1&1&0\\1&0&1&1\\1&1&1&0\end{bmatrix}$$

注意，如果 $\boldsymbol{F}=\boldsymbol{D}\oplus\boldsymbol{E}$，那么当 d_{ij} 和 e_{ij} 都是 0 或者都是 1 时，f_{ij} 是 0。 ■

其次，考虑表 11.2 所给的具有二元运算的集合 $B=\{0,1\}$。该运算早就在不同的背景和不同的符号表示中看到过。在第六章中已经证明 B 是有两个元素的惟一的布尔代数。特别地，B 是具有偏序≤定义为 0≤0，0≤1，1≤1 的一个格。于是读者容易检验如果 a 和 b 是 B 的任意两个元素，那么

$$a\cdot b=a\wedge b \text{ (a 与 b 的最大下界)}$$

因此，表 11.2 恰好是运算 $\wedge$ 重新命名为 $\cdot$ 的表。

表 11.2

$\cdot$	0	1
0	0	0
1	0	1

设 $\boldsymbol{D}=[d_{ij}]$是一个 $m\times p$ 布尔矩阵，$\boldsymbol{E}=[e_{ij}]$是一个 $p\times n$ 布尔矩阵，把**模 2 的布尔积** $\boldsymbol{D}^*\boldsymbol{E}$ 定义为 $m\times n$ 矩阵 $\boldsymbol{F}=[f_{ij}]$，其中

$$f_{ij}=d_{i1}\cdot e_{1j}+d_{i2}\cdot e_{2j}+\cdots+d_{ip}\cdot e_{pj},\qquad 1\leqslant i\leqslant m,1\leqslant j\leqslant n$$

这类乘法在图 11.3 中给予说明。请把该图与 1.5 节的类似图加以比较。

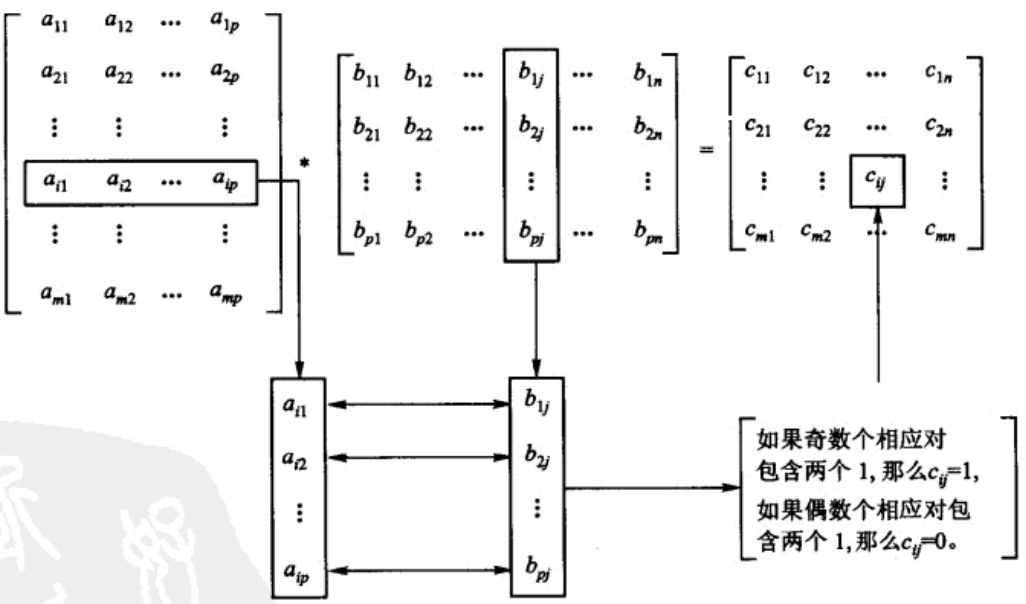

图 11.3

例 10

$$\begin{bmatrix}1&1&0\\0&1&1\end{bmatrix}*\begin{bmatrix}1&0\\1&1\\0&1\end{bmatrix}=\begin{bmatrix}1\cdot1+1\cdot1+0\cdot0&1\cdot0+1\cdot1+0\cdot1\\0\cdot1+1\cdot1+1\cdot0&0\cdot0+1\cdot1+1\cdot1\end{bmatrix}=\begin{bmatrix}0&1\\1&0\end{bmatrix}$$ ■

下面定理的证明留作练习。

定理 4 设 $\boldsymbol{D}$ 和 $\boldsymbol{E}$ 是 $m\times p$ 布尔矩阵，$\boldsymbol{F}$ 是一个 $p\times n$ 布尔矩阵，那么

$$(\boldsymbol{D}\oplus\boldsymbol{E})*\boldsymbol{F}=(\boldsymbol{D}*\boldsymbol{F})\oplus(\boldsymbol{E}*\boldsymbol{F})$$

即：对 $\oplus$ 和*分配性质成立。 ■

现在把元素 $x=x_1x_2\cdots x_n\in B^n$ 看成是 $1\times n$ 矩阵 $[x_1\ \ x_2\ \ \cdots\ \ x_n]$。

定理 5 设 m 和 n 是非负整数且 $m<n$，$r=n-m$，$\boldsymbol{H}$ 是一个 $n\times r$ 布尔矩阵，那么函数 $f_H:B^n\to B^r$ 定义为

$$f_H(x)=x*\boldsymbol{H},\qquad x\in B^n$$

是从群 B^n 到群 B^r 的一个同态。

证明 如果 x 和 y 是 B^n 中的元素，那么

$$\begin{aligned}f_H(x\oplus y)&=(x\oplus y)*\boldsymbol{H}=(x*\boldsymbol{H})\oplus(y*\boldsymbol{H})\qquad\text{由定理 4}\\&=f_H(x)\oplus f_H(y)\end{aligned}$$

因此，f_H 是从 B^n 到 B^r 的一个同态。 ■

推论 1 设 $m, n, r, \boldsymbol{H}$ 和 f_H 如定理 5 所定义，那么

$$N=\left\{x\in B^n\,\middle|\,x*\boldsymbol{H}=\bar{0}\right\}$$

是 B^n 的一个正规子群。

证明 从 9.5 节中的结论得到 N 是同态 f_H 的核，所以它是 B^n 的一个正规子群。 ■

设 $m<n$，$r=n-m$。

$$\boldsymbol{H}=\begin{bmatrix}h_{11}&h_{12}&\cdots&h_{1r}\\h_{21}&h_{22}&\cdots&h_{2r}\\\vdots&\vdots&&\vdots\\h_{m1}&h_{m2}&\cdots&h_{mr}\\1&0&\cdots&0\\0&1&\cdots&0\\\vdots&\vdots&&\vdots\\0&0&\cdots&1\end{bmatrix}\begin{matrix}\\\\\\\\\left.\vphantom{\begin{matrix}1\\0\\\vdots\\0\end{matrix}}\right\}n-m=r\text{行}\end{matrix}$$

是一个 $n\times r$ 布尔矩阵，它的最后 r 行形成 $r\times r$ 单位矩阵，称 $\boldsymbol{H}$ 为**奇偶校验矩阵**。下面用 $\boldsymbol{H}$ 定义一个编码函数 $e_H:B^m\to B^n$。如果 $b=b_1b_2\cdots b_m$，那么设 $x=e_H(b)=b_1b_2\cdots b_mx_1x_2\cdots x_r$，其中

$$\begin{aligned} x_1 &= b_1 \cdot h_{11} + b_2 \cdot h_{21} + \cdots + b_m \cdot h_{m1} \\ x_2 &= b_1 \cdot h_{12} + b_2 \cdot h_{22} + \cdots + b_m \cdot h_{m2} \\ &\vdots \\ x_r &= b_1 \cdot h_{1r} + b_2 \cdot h_{2r} + \cdots + b_m \cdot h_{mr} \end{aligned} \tag{1}$$

定理 6 设 $x = y_1 y_2 \cdots y_m x_1 \cdots x_r \in B^n$，那么 $x * \boldsymbol{H} = \bar{0}$ 当且仅当对某个 $b \in B^m$, $x = e_H(b)$。

证明 假设 $x * \boldsymbol{H} = \bar{0}$，那么

$$\begin{aligned} y_1 \cdot h_{11} + y_2 \cdot h_{21} + \cdots + y_m \cdot h_{m1} + x_1 &= 0 \\ y_1 \cdot h_{12} + y_2 \cdot h_{22} + \cdots + y_m \cdot h_{m2} + x_2 &= 0 \\ &\vdots \\ y_1 h_{1r} + y_2 \cdot h_{2r} + \cdots + y_m \cdot h_{mr} + x_r &= 0 \end{aligned}$$

上面第一个方程具有形式

$$a + x_1 = 0, \qquad a = y_1 \cdot h_{11} + y_2 \cdot h_{21} + \cdots + y_m \cdot h_{m1}$$

把该方程两边同加 a，得到

$$\begin{aligned} & a + (a + x_1) = a + 0 = a \\ & (a + a) + x_1 = a \\ & 0 + x_1 = a \qquad\qquad \text{因为 } a + a = 0 \\ & x_1 = a \end{aligned}$$

对每一行都可如此进行，所以

$$x_i = y_1 \cdot h_{1i} + y_2 \cdot h_{2i} + \cdots + y_m \cdot h_{mi}, \qquad 1 \leqslant i \leqslant r$$

设 $b_1 = y_1, b_2 = y_2, \cdots, b_m = y_m$，则 $x_1, x_2, \cdots, x_r$ 满足方程(1)，因此 $b = b_1 b_2 \cdots b_m \in B^m$ 和 $x = e_H(b)$。

反之，如果 $x = e_H(b)$，那么通过把第 i 个方程两边同加 x_i, $i = 1, 2, \cdots, n$，可以重写方程(1)为

$$\begin{aligned} b_1 \cdot h_{11} + b_2 \cdot h_{21} + \cdots + b_m \cdot h_{m1} + x_1 &= 0 \\ b_1 \cdot h_{12} + b_2 \cdot h_{22} + \cdots + b_m \cdot h_{m2} + x_2 &= 0 \\ &\vdots \\ b_1 h_{1r} + b_2 \cdot h_{2r} + \cdots + b_m \cdot h_{mr} + x_r &= 0 \end{aligned}$$

这就表明 $x * \boldsymbol{H} = \bar{0}$。 ■

推论 2 $e_H(B^m) = \left\{ e_H(b) \middle| b \in B^m \right\}$ 是 B^n 的一个子群。

证明 从推论 1 和

$$e_H(B^m) = \ker(f_H)$$

可以证得结论。因此 e_H 是一个群码。 ■

例 11 设 $m = 2$，$n = 5$，且

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

确定群码 $e_H: B^2 \to B^5$。

解 有 $B^2=\{00,10,01,11\}$,于是

$$e(00)=00x_1x_2x_3$$

其中 x_1, x_2 和 x_3 是由方程(1)决定的，因此

$$x_1=x_2=x_3=0$$
$$e(00)=00000$$

其次

$$e(10)=10x_1x_2x_3$$

用方程(1)并取 $b_1=1$ 和 $b_2=0$，得到

$$x_1=1\cdot1+0\cdot0=1$$
$$x_2=1\cdot1+0\cdot1=1$$
$$x_3=1\cdot0+0\cdot1=0$$

因此 $x_1=1$，$x_2=1$ 和 $x_3=0$，所以 $e(10)=10110$。类似地(验证)有

$$e(01)=01011,\ e(11)=11101$$

■

习 题 11.1

在题 1 和 2 中，求所给字的权。

1. (a) 1011 (b) 0110 (c) 1110

2. (a) 011101 (b) 11111 (c) 010101

3. 考虑(3，4)奇偶校验码，对每个所接收的字，判断是否能检测出一个错误。

(a) 0100 (b) 1100

4. 考虑(3，4)奇偶校验码，对每一个所接收的字，判断是否能检测出一个错误。

(a) 0010 (b) 1001

5. 考虑(6，7)奇偶校验码，对每一个所接收的字，判断是否能检测出一个错误。

(a) 1101010 (b) 1010011

(c) 0011111 (d) 1001101

在第 6 题～第 8 题中，已知 m 的值，用例 3 中的$(m,3m)$解码函数，对每一个所接收的字，判断是否能检测出一个错误。

6. $m=3$

(a) 110111110 (b) 110011011

7. $m=4$

(a) 011010011111 (b) 110110010110

8. m=4

(a) 010010110010 (b) 001001111001

9. ISBN 系统也是错误检测码。在 5.2 节的第 38 题中，它表明能检测出传送中的一个错误，但两个错误不能被检测。证明：在检验码之前的任何相邻数字的对换能被检测出来。

10. 12 位数字的条码使用第 12 位数字作为检验码，它满足条件：在偶数位置数字之和与奇数位置数字之和的三倍关于 0 mod 10 同余。证明：这种条码能检测偶数位置数字的一个错误，但不能检测出偶数位置上的两个错误。

11. 解释如何用$|x \oplus y|$计算 x 和 y 不相同的位置数目？

12. 求 x 和 y 之间的距离。

(a) x=1100010，y=1010001

(b) x=0100110，y=0110010

13. 求 x 和 y 之间的距离。

(a) x=00111001，y=10101001

(b) x=11010010，y=00100111

14. (a) 证明定理 1(a)。

(b) 证明定理 1(b)。

15. 证明定理 1(c)。

16. 求(2，4)编码函数 e 的最短距离。

e(00)=0000
e(10)=0110
e(01)=1011
e(11)=1100

17. 求(3，8)编码函数 e 的最短距离。

e(000)=00000000，e(001)=01110010
e(010)=10011100，e(011)=01110001
e(100)=01100101，e(101)=10110000
e(110)=11110000，e(111)=00001111

18. 考虑(2，6)编码函数 e。

e(00)=000000，e(10)=101010
e(01)=011110，e(11)=111000

(a) 求 e 的最短距离。

(b) e 能检测出多少个错误？

19. 考虑(3，9)编码函数 e。

e(000)=000000000，e(001)=011100101
e(010)=010101000，e(011)=110010001
e(100)=010011010，e(101)=111101011
e(110)=001011000，e(111)=110000111

(a) 求 e 的最短距离。

(b) e 能检测出多少个错误？

20. 验证：(2，5)编码函数 $e: B^2 \to B^5$ 是一个群码，其中

$$e(00)=00000, \quad e(01)=01110$$
$$e(10)=10101, \quad e(11)=11011$$

21. 验证：(3，7)编码函数 e：$B^3 \to B^7$ 是一个群码，其中，

$$e(000)=0000000, \quad e(001)=0010110$$
$$e(010)=0101000, \quad e(011)=0111110$$
$$e(100)=1000101, \quad e(101)=1010011$$
$$e(110)=1101101, \quad e(111)=1111011$$

22. 已知第 20 题中定义的群码，求它的最短距离。

23. 已知第 21 题中定义的群码，求它的最短距离。

24. 计算

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

25. 计算

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

26. 计算

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

27. 计算

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

28. 结构($m \times n$ 布尔矩阵，$\oplus$)有下列哪些性质？

(a) 结合性　　(b) 交换性

(c) 单位元　　(d) 可逆性

29. 结构($n \times n$ 布尔矩阵，$*$)有下列哪些性质？

(a) 结合性　　(b) 交换性

(c) 单位元　　(d) 可逆性

30. 设

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

是一个奇偶校验矩阵，确定(2，5)群码函数 $e_H: B^2 \to B^5$。

31. 设

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

是一个奇偶校验矩阵，确定(3，6)群码 $e_H: B^3 \to B^6$。

32. 证明定理 4。

33. 概括定理 2 的证明结构。

11.2 译码与纠错

考虑一个(m, n)编码函数 e：$B^m \to B^n$。对于 $b \in B^m$，一旦代码字 $x=e(b) \in B^n$ 作为字 x_t 被接收，那么面临识别字 b 是否是源报文的问题。

一个满射函数 d：$B^n \to B^m$ 称为是**与 e 有关的(n, m)译码函数**，如果 $d(x_t)=b' \in B^m$，使得当传输通道没有任何噪声时，$b'=b$，即：$d \circ e = I_{B^m}$，其中 I_{B^m} 是 B^m 上的恒等函数。要求译码函数 d 是满射，从而保证从每个接收的字能够译出 B^m 中所给的字。收到的字是正确的，则译码是正确的，但是如果收到的字是错误的，则译码也许正确也许不正确。

例 1 考虑 11.1 节例 2 中定义的奇偶校验码，定义译码函数 d：$B^{m+1} \to B^m$。如果 $y=y_1y_2\cdots y_my_{m+1} \in B^{m+1}$，那么

$$d(y)=y_1y_2\cdots y_m$$

注意，如果 $b=b_1b_2\cdots b_m \in B^m$，那么

$$(d \circ e)(b)=d(e(b))=b$$

所以 $d \circ e = I_{B^m}$。

对于一个具体的例子，设 m=4，那么得到 d(10010)=1001，d (11001)=1100。 ■

设 e 是一个(m, n)编码函数，d 是与 e 相关的一个(n, m)译码函数。如果无论 $x=e$ (b)是正确

地传输还是有 k 个或更少的错误传输并且接收的是 x_t，都有 $d(x_t)=b$，则称数据对(e, d)**校正 k 个或更少的错误**。因此，从 x_t 能够译出正确的报文 b。

例 2 考虑 11.1 节例 3 中定义的$(m, 3m)$编码函数，现在定义译码函数 $d: B^{3m} \to B^m$。设

$$y = y_1 y_2 \cdots y_m y_{m+1} \cdots y_{2m} y_{2m+1} \cdots y_{3m}$$

那么

$$d(y) = z_1 z_2 \cdots z_m$$

其中

$$z_i = \begin{cases} 1, & \{y_i, y_{i+m}, y_{i+2m}\} \text{ 至少有两个1} \\ 0, & \{y_i, y_{i+m}, y_{i+2m}\} \text{ 有少于两个1} \end{cases}$$

即：在传送的三块数中，译码函数 d 检查每一块的第 i 个数字。在三个块中至少出现两次的数字选作译码的第 i 个数字。对于一个具体的例子，设 m=3，那么

$$e(100)=100100100,\ e(011)=011011011,\ e(001)=001001001$$

现在假设 b=011,那么 $e(011)$=011011011。如果假设传输通道使下面加线的数字出错并且收到字 x_t=011111011，那么，因为三个块中有两个块的第一个数字是 0，所以译出第一个数字是 0。类似地，译出第二个数字是 1，因为在三个块中第二个数字都是 1。最后，出于类似的理由，译出第三个数字也是 1。因此，$d(x_t)$=011，即：译码字是 011，这正好是送出的字。所以，纠正了单一的错误。类似的分析表明如果对如上定义的任意 m 和 d，e 是这个$(m, 3m)$代码，那么(e,d)纠正任意单一的错误。 ■

已知一个(m,n)编码函数 e：$B^m \to B^n$，常常需要确定与 e 相关的一个(n, m)译码函数 d：$B^n \to B^m$。下面讨论一种方法，称作**最大似然方法**，它从一个已知 e 确定一个译码函数 d。

因为 B^m 有 2^m 个元素，所以在 B^n 中存在 2^m 个代码字。首先以一种固定的次序列出代码字：

$$x^{(1)}, x^{(2)}, \cdots, x^{(2m)}$$

如果收到的字是 x_t，那么对于 $1 \leqslant i \leqslant 2^m$，计算 $\delta(x^{(i)}, x_t)$ 并且选取第一个代码字，譬如说 $x^{(s)}$，使得

$$\min_{1 \leqslant i \leqslant 2^m} \{\delta(x^{(i)}, x_t)\} = \delta(x^{(s)}, x_t)$$

即：$x^{(s)}$是所列表中第一个最接近 x_t 的一个代码字。如果 $x^{(s)}=e(b)$，那么用 $d(x_t)=b$ 定义与 e 相关的**最大似然译码函数** d。注意 d 依赖于 $e(B^n)$中代码字所列的特殊顺序。如果代码字是以另一个不同的顺序列出的，那么可以得到与 e 相关的不同的最大似然译码函数 d。

定理 1 假设 e 是一个(m,n)编码函数，d 是与 e 相关的最大似然译码函数，那么(e,d)能纠正 k 个或更少的错误当且仅当 e 的最短距离至少是 $2k+1$。

证明 假设 e 的最短距离至少是 $2k+1$，$b \in B^m$，$x=e(b) \in B^n$。假设 x 传输有 k 个或更少的错误，并且收到的是 x_t。这意味着 $\delta(x,x_t) \leqslant k$。如果 z 是另一个代码字，那么

$$2k+1 \leqslant \delta(x,z) \leqslant \delta(x,x_t) + \delta(x_t,z) \leqslant k + d(x_t,z)$$

因此，$\delta(x_t,z) \geqslant 2k+1-k = k+1$。这意味着 x 是最接近 x_t 的惟一代码字，所以 $d(x_t)=b$。因此，

(e, d)纠正 k 个或更少的错误。

反之，假设代码字之间的最短距离是 $r \leqslant 2k$，并且设 $x=e(b)$和 $x'=e(b')$ 是有 $\delta(x,x')=r$ 的代码字。假设 x' 在定义 d 的代码字列表中先于 x，记 $x=b_1b_2\cdots b_n, x'=b_1'b_2'\cdots b_n'$，那么在 1 到 n 之间恰好有 r 个整数 i 使得 $b_i \neq b_i'$。为简单起见，假设 $b_1 \neq b_1', b_2 \neq b_2', \cdots, b_r \neq b_r'$，但当 $i>r$ 时，$b_i=b_i'$。同理可处理其他情况。

(a) 假设 $r \leqslant k$，如果 x 作为 $x_t=x'$ 传送，那么出现了 $r \leqslant k$ 个错误，但是 $d(x_t)=b'$，所以(e, d)没有纠正这 r 个错误。

(b) 假设 $k+1 \leqslant r \leqslant 2k$，设 $y=b_1'b_2'\cdots b_k'b_{k+1}\cdots b_n$，如果 x 作为 $x_t=y$ 传送，那么 $\delta(x_t,x')=r-k \leqslant k$ 和 $\delta(x_t,x) \geqslant k$。因此，$x'$ 至少是和 x 一样接近 x_t 并且 x' 在代码字列表中先于 x，所以 $d(x_t) \neq b$。于是出现了 k 个错误，这就推出(e, d)没有纠正错误。 ■

例 3 设 e 是 11.1 节例 6 中定义的(3，8)编码函数，d 是与 e 相关的(8，3)最大似然译码函数，(e, d)能纠正多少个错误？

解 因为 e 的最短距离是 3，所以有 $3 \geqslant 2k+1$，即 $k \leqslant 1$。因此，(e, d)能纠正一个错误。 ■

现在讨论关于确定最大似然译码函数的一种简单而有效的方法，该方法与已知群码有关。首先，证明下面的结果。

定理 2 如果 K 是群 G 的一个有限子群，那么 G 中 K 的每一个左陪集与 K 恰好有同样多的元素。

证明 设 aK 是 G 中 K 的左陪集，其中 $a \in G$。考虑函数 f: $K \to aK$ 定义为

$$f(k)=ak, \qquad k \in K$$

可以证明 f 是单射和满射。

为了证明 f 是单射，假设

$$f(k_1)=f(k_2), \qquad k_1, k_2 \in K$$

那么

$$ak_1=ak_2$$

由 9.4 节的定理 2 可知，$k_1=k_2$。因此，f 是单射。

为了证明 f 是满射，设 b 是 aK 中的任意元素，那么对某个 $k \in K$，有 $b=ak$，于是

$$f(k)=ak=b$$

所以 f 是满射。因为 f 是单射和满射，所以 K 与 aK 有相同个数的元素。 ■

设 e: $B^m \to B^n$ 是一个(m, n)编码函数并且是一个群码。因此，B^n 中的代码字集合 N 是 B^n 的一个子群，它的阶是 2^m，如 $N=\{x^{(1)}, x^{(2)}, \cdots, x^{(2^m)}\}$。

假设代码字 $x=e(b)$被传输并且收到的字是 x_t。x_t 的左陪集是

$$x_t \oplus N=\{\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_{2^m}\}$$

其中 $\varepsilon_i=x_t \oplus x^{(i)}$。从 x_t 到代码字 $x^{(i)}$的距离恰好是 ε_i 的权$|\varepsilon_i|$。因此，如果 ε_j 是有最小权的陪集成员，那么 $x^{(j)}$一定是最接近 x_t 的代码字。在这种情况下，$x^{(j)}=\bar{0} \oplus x^{(j)}=x_t \oplus x_t \oplus x^{(j)}=x_t \oplus \varepsilon_j$。

一个有最小权的元素 ε_j 称作**陪集首部**。注意，一个陪集的首部不必是惟一的。

如果 $e: B^m \to B^n$ 是一个群码，那么下面就是关于得到与 e 相关的最大似然译码函数的步骤。

步骤 1：确定 B^n 中 $N=e(B^m)$的所有左陪集。

步骤 2：对于每个陪集，求陪集首部(最小权的字)。步骤 1 和步骤 2 能够用系统化的列表方式实现，稍后描述它。

步骤 3：如果收到字 x_t，那么确定 x_t 属于 N 的陪集。因为 N 是 B^n 的一个正规子群，所以从 9.5 节的定理 3 和定理 4 得到 N 的陪集形成了 B^n 的一个划分，于是 B^n 的每个元素属于 B^n 中 N 的一个且是惟一的陪集。而且，在 B^n 中存在 $2^n/2^m=2^r$ 个 N 的不同陪集。

步骤 4：设 ε 是步骤 3 中确定的陪集的首部，计算 $x=x_t \oplus \varepsilon$。如果 $x=e(b)$，那么设 $d(x_t)=b$，即：x_t 的译码是 b。

为了执行上面的步骤，必须保存 B^n 中 N 的所有陪集的一张完整表。该表通常采用表格形式，表的每一行含有一个陪集。首先确定每一行的陪集首部。然后，当一个字 x_t 被收到时，确定包含它的行，求该行的陪集首部并且把它与 x_t 相加。这就给出了最接近 x_t 的代码字。如果要构造一张更系统化的表，那么可以去掉对这些加法的需求。

在用一个例子说明上述方法之前，提出下面一些注解。设 $N=\left\{x^{(1)},x^{(2)},\cdots,x^{(2^m)}\right\}$，其中 $x^{(1)}$ 是 $\bar{0}$，即 B^n 的单位元。

在上面的译码算法中，步骤 1 和步骤 2 实现如下。首先，在一行的左边从单位元 $\bar{0}$ 开始列出 N 的所有元素。因此，得到

$$\bar{0} \quad x^{(2)} \quad x^{(3)} \quad \cdots \quad x^{(2^m)}$$

该行是陪集$[\bar{0}]$，并且用 $\bar{0}$ 作为它的陪集首部。正由于此，也可以把 $\bar{0}$ 看作 ε_1。现在选择未列在第一行里 B^n 中的任意字 y。把陪集 $y \oplus N$ 中的元素列作第二行。该陪集也有 2^m 个元素。因此，有两行

$$\bar{0} \quad x^{(2)} \quad x^{(3)} \quad \cdots \quad x^{(2^m)}$$

$$y \oplus \bar{0} \quad y \oplus x^{(2)} \quad y \oplus x^{(3)} \quad \cdots \quad y \oplus x^{(2^m)}$$

在陪集 $y \oplus N$ 中，选取一个最小权的元素作为陪集首部，用 $\varepsilon^{(2)}$ 表示。万一最小权元素不止一个，则选取最小权中任意一个元素即可。回顾 9.5 节，因为 $\varepsilon^{(2)} \in y \oplus N$，所以有 $y \oplus N = \varepsilon^{(2)} \oplus N$。这意味着第二行中的每个字能被写作 $\varepsilon^{(2)} \oplus v$，$v \in N$。于是可重写第二行如下：

$$\varepsilon^{(2)} \quad \varepsilon^{(2)} \oplus x^{(2)} \quad \varepsilon^{(2)} \oplus x^{(3)} \quad \cdots \quad \varepsilon^{(2)} \oplus x^{(2^m)}$$

其中 $\varepsilon^{(2)}$ 在最左边的位置上。

其次，选取 B^n 中未列在第一行和第二行中的另一个元素 z 并且形成第三行$(z \oplus x^{(j)})$，$1 \leqslant j \leqslant 2^m$ (B^n 中 N 的另一个陪集)。该行可重写为形如

$$\varepsilon^{(3)} \quad \varepsilon^{(3)} \oplus x^{(2)} \quad \varepsilon^{(3)} \oplus x^{(3)} \quad \cdots \quad \varepsilon^{(3)} \oplus x^{(2^m)}$$

其中 $\varepsilon^{(3)}$ 是这行的陪集首部。

继续这个过程直到列出 B^n 中的所有元素为止。形成的表 11.3 称作**译码表**。注意它包含 2^r 行，每行表示 N 的某个陪集。如果收到字 x_t，那么确定它在表中的位置。如果 x 是包含 x_t 的顶列中 N 的元素，那么 x 是最接近 x_t 的代码字。因此，如果 $x=e(b)$，那么设 $d(x_t)=b$。

表 11.3

$\bar{0}$	$x^{(2)}$	$x^{(3)}$	…	$x^{(2^m-1)}$
$\varepsilon^{(2)}$	$\varepsilon^{(2)}\oplus x^{(2)}$	$\varepsilon^{(2)}\oplus x^{(3)}$	…	$\varepsilon^{(2)}\oplus x^{(2^m-1)}$
⋮	⋮	⋮		⋮
$\varepsilon^{(2^r)}$	$\varepsilon^{(2^r)}\oplus x^{(2)}$	$\varepsilon^{(2^r)}\oplus x^{(3)}$	…	$\varepsilon^{(2^r)}\oplus x^{(2^m-1)}$

例 4 考虑 11.1 节例 7 中定义的(3，6)群码，其中

N={000000，001100，010011，011111，100101，101001，110110，111010}=$\{x^{(1)},x^{(2)},\cdots,x^{(8)}\}$

在例 1 中已定义。对 e 实现上面的译码过程如下。

步骤 1 和步骤 2：确定 B^6 中 N 的所有左陪集，把它作为表的行。对于每行 i，确定陪集首部 ε_i 并且依次重写行：

$$\varepsilon_i,\quad \varepsilon_i\oplus 001100,\quad \varepsilon_i\oplus 010011,\ \cdots,\ \varepsilon_i\oplus 111010$$

结果如表 11.4 所示。

表 11.4

000000	001100	010011	011111	100101	101001	110110	111010
000001	001101	010010	011110	100100	101000	110111	111011
000010	001110	010001	011101	100111	101011	110100	111000
000100	001000	010111	011011	100001	101101	110010	111110
010000	011100	000011	001111	110101	111001	100110	101010
100000	101100	110011	111111	000101	001001	010110	011010
000110	001010	010101	011001	100011	101111	110000	111100
010100	011000	000111	001011	110001	111101	100010	101110

步骤 3 和步骤 4：如果收到字 000101，那么首先通过确定它在译码表中的位置然后译出它；该字出现在第 5 列，即表中下面加线的数字。在第 5 列顶端的字是 100101。因为 e (100)=100101，所以 000101 译码为 100。类似地，如果收到字 010101，那么首先定位它在译码表的第三列上，即表中加两条下划线的数字。在第三列顶端的字是 010011，因为 e (010)=010011。所以 010101 译码为 010。

对于这个例子，提请读者注意，在确定步骤 1 和步骤 2 的译码表时，最后两个陪集的陪集首部存存多个候选字。在第 7 行，选择 00110 作为陪集首部。如果选取 001010 代之，那么第 7 行将会出现下面重新排列的形式

$$001010\quad 001010\oplus 001100\quad \cdots\quad 001010\oplus 111010$$

或

001010 000110 011001 010101 101111 100011 111100 110000

新的译码表如表 11.5 所示。

表 11.5

000000	001100	010011	011111	100101	101001	110110	111010
000001	001101	010010	011110	100100	101000	110111	111011
000010	001110	010001	011101	100111	101011	110100	111000
000100	001000	010111	011011	100001	101101	110010	111110
010000	011100	000011	001111	110101	111001	100110	101010
100000	101100	110011	111111	000101	001001	010110	011010
001010	000110	011001	010101	101111	100011	111100	110000
010100	011000	000111	001011	110001	111101	100010	101110

于是，如果收到字 010101，那么首先确定它在表 11.5 中第 4 列，在该列顶端的字是 011111。因为 $e(011)=011111$，所以 010101 译码为 011。■

假设(m, n)群码是 $e_H: B^m \to B^n$，其中 $\boldsymbol{H}$ 是已知的奇偶校验矩阵。在这种情况下，可以简化上面的译码方法。下面讨论这种情形。

回顾 11.1 节，$r=n-m$。

$$\boldsymbol{H}=\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1r} \\ h_{21} & h_{22} & \cdots & h_{2r} \\ \vdots & \vdots & & \vdots \\ h_{m1} & h_{m2} & \cdots & h_{mr} \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

由 $f_H(x)=x*\boldsymbol{H}$ 定义的函数 $f_H: B^n \to B^r$ 是从群 B^n 到群 B^r 的一个同态。

定理 3 如果 $m, n, r, \boldsymbol{H}$ 以及 f_H 如上定义，那么 f_H 是满射。

证明 设 $b=b_1b_2\cdots b_r$ 是 B^r 中任意元素，且设

$$x=\underbrace{00\cdots0}_{m\text{个}0}b_1b_2\cdots b_r$$

于是得到 $x*\boldsymbol{H}=b$。因此，$f_H(x)=b$，所以 f_H 是满射。■

从 9.5 节的推论 1 得到 B^r 和 B^n/N 是同构的，其中 $N=\ker(f_H)=e_H(B^m)$，同构映射 $g: B^n/N \to B^r$ 定义为

$$g(xN)=f_H(x)=x*\boldsymbol{H}$$

元素 $x*\boldsymbol{H}$ 称为 x 的**校验子**。于是有下面的结果。

定理 4 设 x 和 y 是 B^n 中的元素，那么 x 和 y 属于 B^n 中 N 的相同左陪集当且仅当 $f_H(x)=f_H(y)$，即当且仅当它们有相同的校验子。

证明 从 9.5 节的定理 4 得到 x 和 y 属于 B^n 中 N 的相同左陪集当且仅当 $x\oplus y=(-x)\oplus y\in N$，因为 $N=\ker(f_H)$，所以 $x\oplus y\in N$ 当且仅当

$$f_H(x\oplus y)=\bar{0}_{B^r}$$
$$f_H(x)\oplus f_H(y)=\bar{0}_{B^r}$$
$$f_H(x)=f_H(y)$$

■

在这种情况，先前给出的译码步骤可以修改如下。假设计算每个陪集首部的校验子。如果收到的字是 x_t，那么同样也计算 x_t 的校验子 $f_H(x_t)$。通过 $f_H(x_t)$ 与陪集首部的校验子进行比较，求 x_t 所在的陪集。假设该陪集的陪集首部是 ε，则计算 $x=x_t\oplus\varepsilon$。如果 $x=e(b)$，那么 x_t 译码为 b。因此，为了译码只需陪集首部和它们的校验子。详尽描述新的步骤如下。

步骤 1：确定 B^n 中 $N=e_H(B^m)$ 的所有左陪集。

步骤 2：对每个陪集，求陪集首部并且计算所有首部的校验子。

步骤 3：如果收到 x_t，计算 x_t 的校验子并且求具有相同校验子的陪集首部 ε。那么 $x_t\oplus\varepsilon=x$ 是代码字 $e_H(b)$，所以 $d(x_t)=b$。

对于上面这个过程，不需要保存一张陪集表，并且能够避免计算译码表的工作量。而只要简单地以任意顺序一次列出所有陪集，并且从每个陪集中选择一个陪集首部。然后保存这些陪集首部和它们的校验子的一张表。上面的步骤很容易用这张表实现。

例 5 考虑奇偶校验矩阵

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

和(3，6)群 $e_H: B^3\to B^6$。那么

$$\left.\begin{array}{l}e(000)=000000\\e(001)=001011\\e(010)=010101\\e(011)=011110\\e(100)=100110\\e(101)=101101\\e(110)=110011\\e(111)=111000\end{array}\right\}\text{代码字}$$

因此

N={000000，001011，010101，011110，100110，101101，110011，111000}

实现上面的译码步骤如下。

在表 11.6 中，只给出陪集首部以及它们的校验子。现在假设收到字 001110，则计算 x_t=001110 的校验子，得到 $f_H(x_t)=x_t*\boldsymbol{H}=101$，它是表 11.6 中第一列的第 6 个值。这意味着 x_t 所属的陪集其首部是 ε=010000。计算 $x=x_t\oplus\varepsilon$=001110⊕010000=011110。因为 e (011)=011110，所以 001110 译码为 011。 ■

表 11.6

陪集首部的校验子	陪集首部
000	000000
001	000001
010	000010
011	001000
100	000100
101	010000
110	100000
111	001100

习 题 11.2

1. 设 d 是例 1 中通过设 m 为 3 所定义的(4，3)译码函数。对 B^4 中的字 y，求 $d(y)$。

(a) y=0110　　(b) y=1011

2. 设 d 是例 1 中通过设 m 为 5 所定义的(6，5)译码函数。对 B^6 中的字 y，求 $d(y)$。

(a) y=001101　　(b) y=110100

3. 设 d 是例 2 中定义的(6，2)译码函数。对 B^6 中的字 y，求 $d(y)$。

(a) y=111011　　(b) y=010100

4. 设 d 是像例 2 中那样定义的(9，3)译码函数。对 B^9 中的字 y，求 $d(y)$。

(a) y=101111101　　(b) y=100111100

5. 设 e：$B^2\to B^4$ 是由 e (00)=0000, e (0 1)=1011, e (1 0)=0110, e (11)=1101 所定义的(2，4)编码函数，为 N={0000，1011，0110，1101}构造 B^4 中的一个左陪集表。把陪集首部放在每一行的开始。

6. 设 e 是 11.1 节的第 20 题中定义的(2，5)编码函数，为 $N=e(B^2)$构造 B^4 中的一个左陪集表。把陪集首部放在每一行的开始。

在第 7 题～第 12 题中，设 e 是指定的编码函数，d 是有关的最大似然译码函数，确定(e, d)将纠正的错误的数目。

7. e 是 11.1 节的第 16 题中的编码函数。

8. e 是 11.1 节的第 17 题中的编码函数。

9. e 是 11.1 节的第 18 题中的编码函数。

10. e 是 11.1 节的第 19 题中的编码函数。

11. e 是 11.1 节的第 20 题中的编码函数。

12. e 是 11.1 节的第 21 题中的编码函数。

13. 考虑 11.1 节的第 20 题中所定义的群码，用相关的最大似然译码函数，译出下面各字。

(a) 11110　　(b) 10011　　(c) 10100

14. 考虑(2，4)群码函数 e：$B^2\to B^4$，它定义为

e (00)=0000，　e (01)=0111

e (10)=1001，　e (11)=1111

用相关的最大似然译码函数，译出下面各字。

(a) 0011 (b) 1011 (c) 1111

15. 考虑(3，5)群码编码函数 e：$B^3 \to B^5$，它定义为

$$\begin{aligned} e(000)&=00000, & e(001)&=00110\\ e(010)&=01001, & e(011)&=01111\\ e(100)&=10011, & e(101)&=10101\\ e(110)&=11010, & e(111)&=11100 \end{aligned}$$

用相关的最大似然译码函数，译出下面各字。

(a) 11001 (b) 01010 (c) 00111

16. 考虑(3，6)群码编码函数 e：$B^3 \to B^6$，它定义为

$$\begin{aligned} e(000)&=000000, & e(001)&=000110\\ e(010)&=010010, & e(011)&=010100\\ e(100)&=100101, & e(101)&=100011\\ e(110)&=110111, & e(111)&=110001 \end{aligned}$$

用相关的最大似然译码函数，译出下面各字。

(a) 011110 (b) 101011 (c) 110010

17. 设 G 是一个群，H 是 G 的一个子群。

(a) 证明：如果 g_1 和 g_2 是 G 的元素，则 $g_1H=g_2H$ 或者 $g_1H \cap g_2H=\{\}$。

(b) 利用分题(a)的结果证明：H 的左陪集形成 G 的一个划分。

18. 设 e：$B^m \to B^n$ 是群的编码函数。

(a) 在 B^n 中存在多少个编码字？

(b) 设 $N=e(B^m)$，$|N|$是多少？

(c) 在 B^n 中存在多少个不同的有关 N 的左陪集？

19. 设 e 是第 18 题中定义的群编码函数，给出关于 m 和 n 的条件使得陪集首部具有小于或等于 1 的权。

在第 20 题～第 22 题中，对于已知的奇偶校验矩阵 $\boldsymbol{H}$，求 $N=e_H(B^n)$的陪集首部。

20.

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

21.

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

22.

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

在第 23 题～第 25 题中，计算指定练习中每个陪集首部的校验子。

23. 第 20 题。

24. 第 21 题。

25. 第 22 题。

26. 设

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

是一个奇偶校验矩阵，用相关的最大似然译码函数译出下面各字。

(a) 0101 (b) 1010 (c) 1101

27. 设

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

是一个奇偶校验矩阵，用与 e_H 相关的最大似然译码函数译出下面各字。

(a) 10100 (b) 01101 (c) 11011

28. 设

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

是一个奇偶校验矩阵，用与 e_H 相关的最大似然译码函数译出下面各字。

(a) 011001 (b) 101011 (c) 111010

29. 设 e 是 11.1 节的第 20 题中定义的(2，5)编码函数，d 是与之相关的最大似然译码函数。

(a) 给一个例子来验证(e, d)能纠正一个错误。

(b) 给一个例子验证(e, d)不总能纠正两个错误。

11.3 公钥密码学

1978 年，Ronald Rivest，Adi Shamir 和 Leonard Adelman 发表了一篇文章“一种获取数字签名的方法和公钥密码系统”。在该论文中，作者提出了一种方法，即用一对公开的可得到的整数来发送编码信息。这种方法通常称为 RSA 公钥密码系统。从关于同余的一个结果，即费尔马小定理(9.6 节的定理 3(b))的一个扩展开始说明。

定理 1 假设 p 和 q 是不同的素数，k 是任意一个整数。则

(a) 对任何满足 GCD(a, pq)=1 的整数 a，有

$$a^{k(p-1)(q-1)} \equiv 1 (\text{mod } pq) \tag{1}$$

(b) 对于任何整数 a，有

$$a^{k(p-1)(q-1)+1} \equiv a (\text{mod } pq) \tag{2}$$

证明 (a) 如果 GCD(a, pq)=1，则 a 不能被 p 或 q 整除，即 a 与 p 和 q 都互素。根据费尔马小定理(9.6 节的定理 3(b))，有 $a^{(p-1)} \equiv 1 (\text{mod } p)$，于是

$$a^{k(p-1)(q-1)} \equiv 1^{k(q-1)} = 1 (\text{mod } p)$$

同样，$a^{k(p-1)(q-1)} \equiv 1 (\text{mod } q)$。因此，存在整数 r 和 s 满足

$$a^{k(p-1)(q-1)} = 1 + rp = 1 + sq$$

于是得到 $rp=sq$。同时，由于 q 不能被 p 整除，则 s 必然被 p 整除，假设 $s=pt$，则

$$a^{k(p-1)(q-1)} = 1 + pqt$$

且有

$$a^{k(p-1)(q-1)} \equiv 1 (\text{mod } pq)$$

(b) 如果 a 与 pq 互素，则在式(1)两边乘上 a 就得到结果。不然，则 a 能够被 p 或 q 或两者同时整除。如果 a 被 pq 整除，则式(2)的两边都关于模 pq 与 0 同余，因此它们也是互相同余的。剩下的情况就是 a 只被 p 或 q 之一整除，不失一般性，假设 a 被 p 整除。则 $a = bp^s$，且 $s \geqslant 1$，b 与 pq 互素。在后面将说明 b 必须满足式(2)。

由于 p 与 q 互素，类似于(a)的证明，存在整数 r，使得 $p^{k(p-1)(q-1)} = 1 + rq$。两边乘上 p 得

$$p^{k(p-1)(q-1)+1} \equiv p (\text{mod } pq)$$

于是有

$$(p^s)^{k(p-1)(q-1)+1} = (p^{k(p-1)(q-1)+1})^s \equiv p^s (\text{mod } pq)$$

因此 b 和 p^s 都满足式(2)，故二者相乘的结果 a 也满足式(2)。 ■

例 1 设 p=5 和 q=13。因为 28 与 5×13=65 互素，48=4×12=(5–1)×(13–1)，所以 $28^{48} \equiv 1 (\text{mod } 65)$。 ■

例 2 计算 7^{293} 被 65 除后的余数。

解 使用定理 1(a)，令 p=5，q=13，则 65=pq。有

$$293 = (48 \times 6) + 5$$

且由于 7 和 65 互素，所以

$$7^{293} = (7^{48})^6 \times 7^5 \equiv 7^5 (\text{mod } 65)$$

又由于 $7^3 \equiv 343 \equiv 18(\text{mod } 65)$，则

$$7^5 \equiv 18 \times 49 = 882 \equiv 37(\text{mod } 65)$$

所以 7^{293} 被 65 整除余 37。 ■

现在构造了一个系统，在该系统中用一种公开的方法加密信息，称之为**公钥**。但同时还必须相对确保只有自己才能解密信息。定理 1 对于达到这个目的起着主要作用。首先，注意到任何信息都能够通过许多方法转换为整数的字符串。其中一种方法就是用字母表的字母来表示以 26 为基数的数字。设 A，B，…，Z 分别表示整数 0，1，…，25。于是任何一对字母 $\alpha\beta$ 能够看成是以 26 为基数的数的表达式 $(26\alpha)+\beta$。这样，0 到 675 之间的数字都可以用两个字母的对表示。对于任何信息，当按照两个字母一对进行分组时，都可以在上述范围内用一个整数序列来表示。

例 3 考虑消息 ACT FIRST，把字母分成对，每一对用以 26 为基数所表示的数取代。那么 AC，TF，IR 和 ST 分别变成整数 2，499，225 和 487。如果一个消息有奇数个字母，那么在末尾加一个约定的填充字母，比如 X。也可以把该方法加以变化，用以 26 为基数表示的数来取代三元字母。那么使用的数在 0 到 $25\times 26^2+25\times 26+25=17\,575$ 的范围内。 ■

现在描述一种加密信息的方法。选择两个素数 p 和 q，且令 $m=pq$，$n=(p-1)(q-1)$。现在选择一个与 n 互素的整数 s。“公开”整数 m 和 s (即：使它们可以公开地得到)，并告诉任何想给我们发送秘密信息的人遵循如下步骤：将信息分成字母组 $\alpha\beta$，并将每组用一个数字 $x=(26\alpha)+\beta$ 表示，x 的范围是 0～675。然后将 x 替换为 0～m–1 之间的惟一整数 y，满足 $y \equiv x^s (\text{mod } m)$。这样，只要发来这个最后的整数序列即可。为了使结果惟一，m 必须至少为 675。

解密

由于 s 的选择使得与 n 互素，s 模 n 的余数类 $\bar{s}$ 在环 Z_n 中有一个乘法逆 $\bar{t}$。因此对于某个整数 t，有 $st \equiv 1(\text{mod } n)$ 或者对于某个整数 k 有 $st=1+k(p-1)(q-1)$。可以用欧几里得算法来求 t，如 9.6 节的例 4 所述。如果收到整数 $y \equiv x^s (\text{mod } m)$，计算 $y^t (\text{mod } m)$，并应用定理 1。由于 $m=pq$，通过定理 1(a)，有

$$y^t = x^{st} = x^{1+k(p-1)(q-1)} \equiv x(\text{mod } m)$$

由于 x 不超过 m，得到 $y^t (\text{mod } m) = x$，于是恢复了原来的整数 x。对所有收到的整数进行上述处理，就能够对信息解密了。

例 4 设 $p=19$ 和 $q=37$。因为 $m=pq=703>675$，所以能用 RSA 方法加密两个字母一组的报文。这里 $n=18\times 36=648$。选取与 648 互素的数 $s=25$。现在把整数 703 和 25 作为公钥公开。如果某人要传送消息 GO，她首先计算 $6\times 26+14=170$，然后计算 $170^{25} (\text{mod } 703)$。注意

$170^2=28900\equiv 77(\text{mod } 703)$。所以

$$170^4\equiv 77^2=305(\text{mod } 703)$$
$$170^8\equiv 305^2=229(\text{mod } 703)$$
$$170^{16}\equiv 229^2=419(\text{mod } 703)$$

从而得到

$$170^{25}=170^{16}170^8170\equiv 419\cdot 229\cdot 170=16\ 311\ 670\equiv 664(\text{mod } 703)$$

所以它传送 664。

为了解密消息，首先求 t。使用欧几里得算法，计算

$$648=25\times 25+23$$
$$25=1\times 23+2$$
$$23=11\times 2+1$$

因此

$$\begin{aligned}1&=23-11\times 2=23-11\times(25-23)\\&=12\times 23-11\times 25=12\times(648-25\times 25)-11\times 25\\&=12\times 648-311\times 25\end{aligned}$$

故 $t=-311\equiv 337(\text{mod } 648)$。

现在计算 $664^{337}(\text{mod } 703)$。对 664 经过一系列上面使用的那些方法进行计算，表明 $664^{337}\equiv 170(\text{mod } 703)$。因为 $6\times 26+14=170$，所以能恢复原来的消息 GO。 ■

安全性

在讨论 Bacon 编码的时候(1.4 节)，注意到这种编码方法通过对普通语言中字母出现的频率进行分析的方法来攻击是存在漏洞的。如果用本节的公钥方法，对二元字或三元字的字母组合进行编码，那么通过频率分析来实现攻击要困难得多。但仍然可以找到其他方法对公钥密码系统进行攻击。

为了对信息解密，人们必须知道 t，这意味着必须知道 n。这又要求知道 p 和 q。所以，问题是要对数 m 进行因式分解，已知 m 是两个素数的积。在 1.4 节已经证明如果用比 $\sqrt{m}$ 要小的所有素数去除 n，通过尝试和误差的方法就能求出 m 的素数因子。在例 4 中，m=703,它的平方根比 27 小。因此要求出它的素数因子至多用 2，3，5，7，11，13，17，19 和 23 这 9 个除数去除就可以了。在实际应用中，人们选择 p 和 q 都是 100 位的十进制数，即 10^{100} 数量级的数，那么 m 是 10^{200} 数量级的数。一个著名的关于素数的定理，称作近似素数定理，指出小于或等于一个整数 m 的素数的个数约为 $\dfrac{m}{\ln(m)}$，并且随着 m 的增大，近似效果更好。因此，比 $\sqrt{m}=10^{100}$ 小的素数的个数约为

$$\frac{10^{100}}{\ln(10^{100})}=\frac{10^{100}}{100\ln 10}\approx\frac{10^{100}}{230}>10^{97}$$

现在，最快的计算机的速度为每秒 360 000 亿次运算。用这台计算机计算所需的除法次数需要花10^{83}秒的时间，相当于10^{67}亿年。同样，*即使知道了全部这些结果*，还需要如此巨大的世界上最大的硬盘来保存这些素数分解的结果。

大整数的因式分解的难度提供了某种程度上的安全性。即使如此，如果有额外的信息泄漏，信息仍可能被破解。例如，如果知道 $n=(p-1)(q-1)$，就能求出因式分解。根据 p 和 q 都是二次方程式

$$0=(x-p)(x-q)=x^2-(p+q)x+pq=x^2-(p+q)x+m$$

的解，另外，有

$$n=(p-1)(q-1)=pq-(p+q)+1=m-(p+q)+1$$

于是 $(p+q)=m-n+1$。这样可以通过求解方程 $0=x^2+(n-m-1)x+m$ 得到 p 和 q。

编码的效率、错误检测和纠错以及它的安全性都是数学研究中的活跃领域。本书仅仅给出了一些基本的概念、原理和方法。

习 题 11.3

1. 验证 $12^{704}\equiv 1(\mathrm{mod}\ 391)$。
2. 验证 $10^{577}\equiv 10(\mathrm{mod}\ 221)$。

在第 3 题～第 6 题中，对已知的值，当 a^k 用 c 除时，求余数。

3. $a=9$，$k=199$，$c=221$。
4. $a=17$，$k=1\ 123$，$c=1\ 189$。
5. $a=23$，$k=3\ 750$，$c=3\ 869$。
6. $a=12$，$k=1\ 540$，$c=1\ 649$。
7. 设 $p=23$，$q=41$。

(a) 计算 $m=pq$，$n=(p-1)(q-1)$。

(b) 设 $s=41$，求 t 使得 $st\equiv 1(\mathrm{mod}\ n)$。

8. 使用第 7 题中的 m 和 s 以及字母对，应用 RSA 方法加密报文 BEAR。
9. 使用第 7 题中的 m 和 s 以及字母对，应用 RSA 方法解密消息 371，640。

在第 10 题～第 12 题中，使用 RSA 方法、字母对以及公钥 $m=779$，$s=49$。

10. 加密报文 STOP。
11. 加密报文 NO。
12. 加密报文 EXIT。

在第 13 题～第 15 题中，使用 RSA 方法、三个字母以及公钥 $m=19\ 781$，$s=19$。

13. 加密报文 RUN。
14. 加密报文 YES。
15. 加密报文 END。
16. 已公开公钥 m=779，s=49，假设你发现该密码系统的 n=720，求 p 和 q。
17. 已公开公钥 m=19 781，s=19，假设你发现该密码系统的 n=19 500，求 p 和 q。
18. 使用第 16 题中的信息解密消息 142，525。
19. 使用第 17 题中的信息解密消息 14 032。

证明知识小结

本章中的证明密切依赖于前面几章的结果。本书所建立的许多概念被应用到本章编码和译码问题中。在 11.1 节，已指出证明两个数相等与证明两个集合相等是类似的。因此，对于任意有反对称性质的关系，类似的证明可以发展，就像“比……少”和“是……的一个子集”一样。

在 11.2 节的定理 2 中，为了证明两个集合有同样多的元素，使用了一个单射和满射函数来“匹配”两个集合中的元素。如果其中有一个集合的基数是已知的，那么这也是一种能用来解计数问题的技巧。

主要概念复习

- 报文：从有限个字母表中得到的字符的有限序列。
- 字：0 和 1 的序列。
- (m,n)编码函数：单射函数 e：$B^m \to B^n$，$m<n$。
- 代码字：Ran(e)中的元素。
- x 的权，$|x|$：x 中 1 的个数。
- 奇偶校验码：见 11.1 节。
- x 与 y 之间的 Hamming 距离，$\delta(x,y):|x \oplus y|$。
- 定理(距离函数性质)：设 x, y 和 z 是 B^m 的元素，那么
  (a) $\delta(x,y)=\delta(y,x)$。
  (b) $\delta(x,y) \geqslant 0$。
  (c) $\delta(x,y)=0$ 当且仅当 $x=y$。
  (d) $\delta(x,y) \leqslant \delta(x,z)+\delta(z,y)$。
- (m,n)编码函数的最短距离：所有不同对代码字之间的最短距离。
- 定理：一个(m,n)编码函数 e：$B^m \to B^n$ 能够检测 k 个或更少错误当且仅当它的最短距离至少是 k+1。
- 群码：(m,n)编码函数 e：$B^m \to B^n$ 使得 $e(B^m)=\{e(b)|b \in B^m\}$ 是 B^n 的一个子群。

- 定理：群码的最短距离是非零代码字的最小权。
- 布尔矩阵 $\boldsymbol{D}$ 与 $\boldsymbol{E}$ 的模 2 和，$\boldsymbol{D}\oplus\boldsymbol{E}$：见 11.1 节。
- 布尔矩阵 $\boldsymbol{D}$ 与 $\boldsymbol{E}$ 的模 2 布尔积，$\boldsymbol{D}^*\boldsymbol{E}$：见 11.1 节。
- 定理：设 m 和 n 是非负整数，$m<n$，$r=n-m$，$\boldsymbol{H}$ 是一个 $n\times r$ 布尔矩阵，那么函数 $f_H: B^n\to B^r$ 是从群 B^n 到群 B^r 的一个同态，其中 f_H 定义为
$$f_H(x)=x^*\boldsymbol{H},\quad x\in B^n$$
- 对应于奇偶校验矩阵 $\boldsymbol{H}$ 的群码 e_H：见 11.1 节。
- (n,m)译码函数：见 11.2 节。
- 与 e 相关的最大似然译码函数：见 11.2 节。
- 定理：假设 e 是一个(m,n)编码函数并且 d 是与 e 相关的最大似然译码函数，那么(e,d)能纠正 k 个或更少错误当且仅当最短距离至少是 $2k+1$。
- 群码的译码过程：见 11.2 节。
- 由奇偶校验矩阵给出的群码的译码过程：见 11.2 节。
- RSA 公钥密码系统：见 11.3 节。
- 定理：假设 p 和 q 是不同的素数，k 是任意一个整数，那么
  (a) 对于任意整数 a 且 $\mathrm{GCD}(a,pq)=1$，$a^{k(p-1)(q-1)}\equiv 1(\mathrm{mod}\,pq)$。
  (b) 对于任意整数 a，$a^{k(p-1)(q-1)+1}\equiv a\,(\mathrm{mod}\,pq)$。

回顾问题

1. 一个(m,n)编码函数的最短距离与它能检测到的错误数之间有什么关系？
2. 为什么 $e(B^m)$的左陪集对于形成 B^n 的一个划分是很重要的？
3. 本章中的最大似然译码函数涉及什么内容？
4. 编码一个消息的三条基本原则是什么？

第十一章自测题

1. 考虑(3，4)奇偶校验码，对每一个所接收的字，判断是否能检测出一个错误。

(a) 1101 (b) 1010 (c) 1111 (d) 0011

2. 考虑 $m=4$ 的$(m,3m)$译码函数。对每一个接收字，判断是否能检测出一个错误。

(a) 001100100011 (b) 110111001101 (c) 010111010011

3. 设 e 是由下面定义的(3，5)编码函数。

$e(000)=00000$，　$e(100)=01010$
$e(001)=11110$，　$e(101)=10100$
$e(010)=01101$，　$e(110)=00111$

$$e(011)=10011, \qquad e(111)=11001$$

则 e 能检测出多少个错误？

4. 证明：第 3 题中的(3，5)编码函数是一个群码。

5. 设 e 是第 3 题中定义的编码函数，d 是与它相关的最大似然译码函数，求(e, d)的纠错数目。

6. 设

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

是一个奇偶校验矩阵，用与 e_H 相关的最大似然译码函数译出 0110。

7. 当 58^{1226} 用 91 除时，计算其余数。

8. 用 RSA 方法，字母对和公钥 m=91, s=25 来加密报文 LAST。

编码练习

对于下面各题，用伪码(见附录 A 描述)或你所熟悉的程序设计语言写出符合要求的程序或子程序，通过手工记录或计算机运行来检验你的编码。

1. 写一个函数求 B^n 中的字的权。

2. 写一个子程序，计算 B^n 中两个字之间的 Hamming 距离。

3. 设 $\boldsymbol{M}$ 和 $\boldsymbol{N}$ 都是 $n\times n$ 布尔矩阵，写一个程序已知 $\boldsymbol{M}$ 和 $\boldsymbol{N}$，返回它们的模 2 布尔积。

4. 写一个子程序模拟 11.1 节例 3 中描述的$(m, 3m)$编码函数 e：$B^m \to B^{3m}$。

5. 像 11.2 节的例 2 所描述的那样，为第 4 题中的编码函数写一个模拟译码函数 d 的子程序。