ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC KHOA HỌC

NGUYỄN TRỌNG NAM

LÝ THUYẾT ĐỒNG DƯ VÀ ỨNG DỤNG TRONG MÃ SỬA SAI

LUẬN VĂN THẠC SĨ TOÁN HỌC

THÁI NGUYÊN - 2009

ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC KHOA HỌC

NGUYỄN TRỌNG NAM

LÝ THUYẾT ĐỒNG DƯ VÀ ỨNG DỤNG TRONG MÃ SỬA SAI

Chuyên ngành: TOÁN SƠ CẤP

Mã số: 60.46.40

LUẬN VĂN THẠC SĨ TOÁN HỌC

Người hướng dẫn khoa học: PGS.TS TẠ DUY PHƯỢNG

THÁI NGUYÊN - 2009

MỤC LỤC

LỜI NÓI ĐẦU

Có thể nói, số học, lý thuyết số là một trong những kiến thức toán học lâu đời nhất. Từ trước tới nay, người ta thường coi lý thuyết số như một lĩnh vực đẹp, nhưng thuần túy lý thuyết, của toán học. Với sự phát triển của khoa học máy tính và công nghệ thông tin, lý thuyết số đã đóng góp những ứng dụng thực tế bất ngờ và quan trọng, đặc biệt trong lĩnh vực mã hóa thông tin.

Nhiều khía cạnh khác nhau của mã hóa thông tin được các nhà toán học và tin học quan tâm. Thường thường thông tin được mã hóa qua dãy các chữ số trong hệ đếm cơ số 2, cơ số 10, hoặc cơ số p nào đó. Trong quá trình truyền tin hoặc nhận tin, vì nhiều lý do, thông tin có thể bị sai lệch. Thí dụ, một tin nhắn được mã hóa trong cơ số 2 khi truyền đi bị sai một lỗi (lỗi đơn) thì điều này có nghĩa là chữ số 1 tại vị trí nào đó đã bị đổi thành chữ số 0 hoặc ngược lại. Một trong những vấn đề cần giải quyết là phát hiện ra các lỗi sai và sửa chúng.

Vì yêu cầu thực tiễn đó, lý thuyết mã sửa sai đã ra đời, phát triển và có những ứng dụng thực tiễn quan trọng. Để xây dựng lý thuyết mã sửa sai, các nhà toán học và khoa học máy tính đã sử dụng nhiều thành tựu của toán học hiện đại (số học, toán rời rạc, đại số tuyến tính,...,) đặc biệt là số học trên tập số nguyên, trong đó có lý thuyết đồng dư.

Luận văn này có mục đích tìm hiểu và trình bày những kiến thức cơ bản nhất của lý thuyết mã sửa sai trên cơ sở lý thuyết đồng dư và lý thuyết trường hữu hạn.

Luận văn gồm hai chương.

Chương 1 trình bày các kiến thức cơ bản nhất của lý thuyết đồng dư và lý thuyết trường hữu hạn, chủ yếu dựa theo tài liệu [2], có tham khảo thêm các tài liệu [4] và [6].

Chương 2 trình bày một số vấn đề cơ bản của mã sửa sai: khoảng cách Hamming; phát hiện và sửa lỗi; các thuật toán giải mã; mã hoàn hảo; mã tuyến tính và ma trận kiểm tra, xây dựng mã tuyến tính,...

Nội dung của Chương 2 trình bày chủ yếu dựa theo tài liệu [6], có tham khảo thêm các tài liệu [1] và [7]. Ngoài ra, chúng tôi cũng quan tâm đến khía cạnh thực tế của vấn đề: mã vạch, mã hàng hóa, mã sách tiêu chuẩn quốc tế,.... Chúng tôi cũng cố gắng tìm hiểu, tuy chưa được đầy đủ, các mã hàng hóa, mã văn hóa phẩm của Việt Nam và kiểm nghiệm các tiêu chuẩn giải mã cho các ví dụ cụ thể của các mã này.

Luận văn được hoàn thành dưới sự hướng dẫn khoa học của PGS TS Tạ Duy Phượng. Xin được tỏ lòng cám ơn chân thành nhất tới Thầy.

Tác giả xin cám ơn chân thành tới Trường Đại học Khoa học Thái Nguyên, nơi tác giả đã nhận được một học vấn sau đại học căn bản.

Và cuối cùng, xin cám ơn gia đình, bạn bè, đồng nghiệp đã cảm thông, ủng hộ và giúp đỡ trong suốt thời gian tác giả học Cao học và viết luận văn.

Hà Nội, ngày 19 tháng 9 năm 2009

Tác giả

Nguyễn Trọng Nam

Chương 1
LÝ THUYẾT ĐỒNG DƯ

§1. Quan hệ đồng dư

1.1. Định nghĩa đồng dư

Kí hiệu $\mathbb{Z}$ là tập hợp các số nguyên.

Định nghĩa

Cho m là một số nguyên dương, a và b là hai số nguyên. Ta nói a và b *đồng dư* với nhau theo môđun m nếu trong phép chia a và b cho m ta được cùng một số dư, nghĩa là có các số nguyên q_1, q_2, r với $0 \le r < m$ sao cho

$$a = mq_1 + r \text{ và } b = mq_2 + r.$$

Khi a và b đồng dư với nhau theo môđun m, ta viết $a \equiv b \pmod{m}$.

Nếu a không đồng dư với b theo môđun m thì ta viết $a \not\equiv b \pmod{m}$.

Định lý

Các mệnh đề sau là tương đương.

i. a và b đồng dư với nhau theo môđun m;

ii. a – b chia hết cho m (kí hiệu là $m \mid (a-b)$);

iii. Tồn tại số nguyên t sao cho $a = b+mt$.

Chứng minh

i $\Rightarrow$ ii. Ta có $a \equiv b \pmod{m} \Leftrightarrow a = mq_1 + r$, $b = mq_2 + r$ với $q_1, q_2, r \in \mathbb{Z}$, $0 \le r < m$. Suy ra $a - b = m(q_1 - q_2)$. Do $q_1 - q_2 \in \mathbb{Z}$ nên $m \mid (a-b)$.

ii $\Rightarrow$ iii. Giả sử $m \mid (a-b)$. Khi ấy tồn tại số $t \in \mathbb{Z}$ sao cho $a - b = mt$, tức là $a = b + mt$.

iii $\Rightarrow$ i. Giả sử có số $t \in \mathbb{Z}$ sao cho $a = b + mt$. Gọi r là số dư trong phép chia a cho m, nghĩa là $a = mq_1 + r$ với $q_1, r \in \mathbb{Z}$, $0 \le r < m$. Khi ấy:

$b+mt=a=mq_1+r$ hay $b=m(q_1-t)+r$, trong đó $q_1-t\in\mathbb{Z}$, $0\le r<m$.

Chứng tỏ số dư trong phép chia b cho m cũng là r, tức là $a\equiv b(\mathrm{mod}\ m)$.

1.2. Các tính chất của quan hệ đồng dư

a. Quan hệ đồng dư là một quan hệ tương đương trên tập $\mathbb{Z}$:

i. Với mọi $a\in\mathbb{Z}$: $a\equiv a(\mathrm{mod}\ m)$;

ii. Với mọi $a,b\in\mathbb{Z}$: $a\equiv b(\mathrm{mod}\ m)$ khi và chỉ khi $b\equiv a(\mathrm{mod}\ m)$;

iii. Với mọi $a,b,c\in\mathbb{Z}$: $a\equiv b(\mathrm{mod}\ m)$, $b\equiv c(\mathrm{mod}\ m)$ suy ra $a\equiv c(\mathrm{mod}\ m)$.

Chứng minh

i. Vì $a-a$ chia hết cho m nên $a\equiv a(\mathrm{mod}\ m)$.

ii. Từ $a\equiv b(\mathrm{mod}\ m)$ ta có $m\mid(a-b)$. Do đó $m\mid(b-a)\Rightarrow b\equiv a(\mathrm{mod}\ m)$.

iii. Ta có $a\equiv b(\mathrm{mod}\ m)$ và $b\equiv c(\mathrm{mod}\ m)$ nên $m\mid(a-b)$ và $m\mid(b-c)$.

Khi đó $m\mid((a-b)+(b-c))$ hay $m\mid(a-c)$. Vậy $a\equiv c(\mathrm{mod}\ m)$.

b. Ta có thể cộng hoặc trừ từng vế của nhiều đồng dư thức theo cùng một môđun. Cụ thể là, nếu $a_1\equiv b_1(\mathrm{mod}\ m)$ và $a_2\equiv b_2(\mathrm{mod}\ m)$ thì ta có:

$$a_1\pm a_2\equiv b_1\pm b_2(\mathrm{mod}\ m).$$

Chứng minh

Từ $a_1\equiv b_1(\mathrm{mod}\ m)$, $a_2\equiv b_2(\mathrm{mod}\ m)$ suy ra tồn tại $t_1,t_2\in\mathbb{Z}$ sao cho $a_1=b_1+mt_1$, $a_2=b_2+mt_2$. Do đó $a_1\pm a_2\equiv b_1\pm b_2+m(t_1\pm t_2)$ với $t_1\pm t_2\in\mathbb{Z}$. Vậy $a_1\pm a_2\equiv b_1\pm b_2(\mathrm{mod}\ m)$.

c. Ta có thể nhân từng vế của nhiều đồng dư thức theo cùng một môđun. Cụ thể là, nếu $a_1\equiv b_1(\mathrm{mod}\ m)$, $a_2\equiv b_2(\mathrm{mod}\ m)$ thì $a_1a_2\equiv b_1b_2(\mathrm{mod}\ m)$.

Chứng minh

Từ $a_1 \equiv b_1 (\text{mod } m)$, $a_2 \equiv b_2 (\text{mod } m)$ suy ra tồn tại $t_1, t_2 \in \mathbb{Z}$ sao cho $a_1 = b_1 + mt_1$, $a_2 = b_2 + mt_2$.

Do đó $a_1 a_2 = b_1 b_2 + m(b_2 t_1 + b_1 t_2 + mt_1 t_2)$, $b_2 t_1 + b_1 t_2 + mt_1 t_2 \in \mathbb{Z}$.

Vậy $a_1 a_2 - b_1 b_2$ chia hết cho m hay $a_1 a_2 \equiv b_1 b_2 (\text{mod } m)$.

d. Hệ quả

1. $a \equiv b (\text{mod } m)$ khi và chỉ khi $a \pm c \equiv b \pm c (\text{mod } m)$.

Thật vậy, ta có $a \equiv b (\text{mod } m)$ và $c \equiv c (\text{mod } m)$.

Vậy $a \pm c \equiv b \pm c (\text{mod } m)$.

2. $a + c \equiv b (\text{mod } m)$ khi và chỉ khi $a \equiv (b - c) (\text{mod } m)$.

Thật vậy, ta có $a \equiv b (\text{mod } m)$, $c \equiv c (\text{mod } m)$. Vậy $a \equiv (b - c) (\text{mod } m)$.

3. $a \equiv b (\text{mod } m)$ khi và chỉ khi $a \pm km \equiv b (\text{mod } m)$ với mọi $k \in \mathbb{Z}$.

Thật vậy, $a \equiv b (\text{mod } m)$, $km \equiv 0 (\text{mod } m)$. Vậy $a \pm km \equiv b (\text{mod } m)$.

4. $a \equiv b (\text{mod } m)$ khi và chỉ khi $ac \equiv bc (\text{mod } m)$.

Ta có $a \equiv b (\text{mod } m)$, $c \equiv c (\text{mod } m)$. Vậy $ac \equiv bc (\text{mod } m)$.

5. $a \equiv b (\text{mod } m) \Rightarrow a^n \equiv b^n (\text{mod } m)$ $\forall n \in \mathbb{Z}$, $n > 0$.

Ta có $a \equiv b (\text{mod } m)$; $a \equiv b (\text{mod } m)$; ...; $a \equiv b (\text{mod } m)$

Suy ra $a^n \equiv b^n (\text{mod } m)$.

6. Giả sử $f(x)$ là một đa thức với hệ số nguyên và $\alpha \equiv \beta (\text{mod } m)$. Khi ấy

$$f(\alpha) \equiv f(\beta) (\text{mod } m)$$

Đặc biệt, nếu $f(\alpha) \equiv 0 (\text{mod } m)$ thì $f(\alpha + km) \equiv 0 (\text{mod } m)$ với mọi $k \in \mathbb{Z}$.

Chứng minh

Thật vậy, giả sử $f(x) = a_0 + a_1x + ... + a_nx_n$. Từ giả thiết $\alpha \equiv \beta \pmod m$ suy ra $a_i\alpha^i \equiv a_i\beta^i \pmod m$, $i = 1, 2, ..., n$. Do đó

$$a_0 + a_1\alpha + ... + a_n\alpha^n \equiv a_0 + a_1\beta + ... + a_n\beta^n \pmod m,$$

nghĩa là $f(\alpha) \equiv f(\beta) \pmod m$.

Đặc biệt, vì $\alpha \equiv (\alpha + km) \pmod m$ $\forall k \in \mathbb{Z}$ nên $f(\alpha) \equiv f(\alpha + km) \pmod m$.

Nhưng $f(\alpha) \equiv 0 \pmod m$ nên ta có $f(\alpha + km) \equiv 0 \pmod m$ với mọi $k \in \mathbb{Z}$.

e. Ta có thể chia hai vế của một đồng dư thức cho một ước chung của chúng nguyên tố với môđun m:

$$ac \equiv bc \pmod m \text{ và } UCLN(c, m) = 1 \Rightarrow a \equiv b \pmod m.$$

Chứng minh

Ta có $ac \equiv bc \pmod m \Rightarrow m|(ac - bc)$ hay $m|c(a - b)$. Nhưng $(m, c) = 1$ nên ta có $m|(a - b) \Rightarrow a \equiv b \pmod m$.

f. Có thể chia hai vế và môđun của một đồng dư thức cho một ước chung dương của chúng:

$$a \equiv b \pmod m,\ 0 < \delta \in \mathbb{Z},\ \delta | UCLN(a, b, m) \Rightarrow \frac{a}{\delta} \equiv \frac{b}{\delta} \left(\mod \frac{m}{\delta} \right).$$

Chứng minh

Từ giả thiết $\delta|(a, b, m)$, ta đặt $a = \delta a_1$, $b = \delta b_1$, $m = \delta m_1$ với a_1, b_1, $m_1 \in \mathbb{Z}$, $m_1 > 0$. Mặt khác, $a \equiv b \pmod m \Rightarrow a = b + mt$, $t \in \mathbb{Z}$. Ta có:

$$\delta a_1 = \delta b_1 + \delta m_1 \Rightarrow a_1 = b_1 + m_1 t \Rightarrow a_1 \equiv b_1 \pmod{m_1} \text{ hay } \frac{a}{\delta} \equiv \frac{b}{\delta} \left(\mod \frac{m}{\delta} \right).$$

g. Nếu hai số đồng dư với nhau theo một môđun thì chúng cũng đồng dư theo môđun là ước của môđun ấy:

$$a \equiv b \pmod m,\ \delta | m,\ \delta > 0 \Rightarrow a \equiv b \pmod m.$$

Chứng minh

Từ $a \equiv b \pmod{m} \Rightarrow m|(a - b)$, mà $\delta|m \Rightarrow \delta|(a - b) \Rightarrow a \equiv b \pmod{\delta}$.

h. Nếu hai số đồng dư với nhau theo nhiều môđun thì chúng đồng dư với nhau theo môđun là bội chung nhỏ nhất của các môđun ấy:

$a \equiv b \pmod{m_i},\ i = 1,..., k \Rightarrow a \equiv b \pmod{m}$ với $m = \text{BCNN}(m_1, m_2, \ldots, m_k)$.

i. Nếu hai số đồng dư với nhau theo một môđun thì chúng có cùng UCLN với môđun ấy:

$a \equiv b \pmod{m}$ thì $UCLN(a, m) = UCLN(b, m)$.

§2. Thặng dư

2.1. Tập các lớp thặng dư

Cho m là số nguyên dương. Theo tính chất của đồng dư thức, quan hệ đồng dư là quan hệ tương đương trong tập trong tập số nguyên $\mathbb{Z}$. Ta nói, các số nguyên a và b cùng thuộc lớp tương đương A nếu chúng đồng dư với nhau. Như vậy, $\mathbb{Z}$ có thể được phân thành các lớp theo quan hệ tương đương. Nói cách khác, tồn tại tập thương $\mathbb{Z}$ trên quan hệ tương đương. Ta có

Định nghĩa

Tập thương của tập hợp số nguyên $\mathbb{Z}$ trên quan hệ đồng dư theo môđun m được gọi là *tập hợp các lớp thặng dư* môđun m, kí hiệu là $\mathbb{Z}_m$.

Mỗi phần tử A của $\mathbb{Z}_m$ được gọi là một *lớp thặng dư* môđun m.

Từ định nghĩa, hai lớp thặng dư môđun m hoặc bằng nhau hoặc không giao nhau và $\mathbb{Z}_m$ là hợp của tất cả các lớp thặng dư môđun m rời nhau.

Giả sử $A \in \mathbb{Z}_m$ và $a \in A$, Khi ấy $A = \{x \in \mathbb{Z} : x \equiv a \pmod{m}\}$.

Phần tử a được gọi là *đại diện* của lớp thặng dư A và cũng được gọi là *một thặng dư* môđun m.

Nhiều khi ta cũng viết $A = \overline{a} = \{x \in \mathbb{Z} : x \equiv a \pmod{m}\}$ để thể hiện a là đại diện cho lớp thặng dư $A = \overline{a}$.

2.2. Các tính chất của lớp thặng dư

Tính chất 1

Tập $\mathbb{Z}_m$ có m phần tử.

Chứng minh

Xét các lớp thặng dư môđun m: $\bar{0}, \bar{1}, ..., \overline{m-1}$. Ta sẽ chứng minh chúng gồm: m lớp phân biệt của $\mathbb{Z}_m$ và mỗi lớp $\bar{x}$ của $\mathbb{Z}_m$ phải trùng với một trong m lớp đã nêu, do đó $\mathbb{Z}_m = \{\bar{0}, \bar{1}, ..., \overline{m-1}\}$.

Thật vậy, với $i \neq j$ $(0 \leq i,\ j \leq m-1)$ thì $0 < |i-j| \leq m-1$ nên $i - j \not\equiv 0$, nghĩa là $i \not\equiv j \pmod m$ hay $\bar{i} \neq \bar{j} \pmod m$. Như vậy $\bar{0}, \bar{1}, ..., \overline{m-1}$ là m lớp thặng dư phân biệt, chúng tạo nên một tập con X gồm m phần tử của $\mathbb{Z}_m$.

Giả sử $\bar{x} \in \mathbb{Z}_m$ và $x = mq + i$, $i, q \in \mathbb{Z}$, $0 \leq i \leq m-1$ thì $x \equiv i \pmod m$ nên $\bar{x} = \bar{i} \in \{\bar{0}, \bar{1}, ..., \overline{m-1}\} = X$. Vậy $\mathbb{Z}_m = X = \{\bar{0}, \bar{1}, ..., \overline{m-1}\}$ có m phần tử.

Tính chất 2

Mỗi lớp phần tử của $\mathbb{Z}_m$ là tập hợp của k phần tử phân biệt của $\mathbb{Z}_{km}$, $k > 1$.

Chứng minh

Giả sử $A = \bar{a} \in \mathbb{Z}_m$. Ta sẽ chứng minh A là hợp của k phần tử ($k > 1$) đôi một không giao nhau của $\mathbb{Z}_{km}$ xác định như sau:

$A_0 \equiv \bar{a} \pmod{km}$, $A_1 = \overline{a+m} \pmod{km}$, ..., $A_{k-1} = \overline{a+(k-1)m} \pmod{km}$.

Trước hết, với $i \neq j$, $(0 \leq i, j \leq k\text{-}1)$ ta có $0 < |(a+im) - (a-jm)| < km$ nên $a + im \not\equiv a + jm \pmod{km}$. Suy ra $A_i \neq A_j$. Do đó $A_i \cap A_j = \varnothing$.

Ta có $A = \bigcup_{i=0}^{k-1} A_i$.

Thật vậy, giả sử $x \in A = \bar{a} \pmod m$. Ta có $x \equiv a \pmod m$ nên $x = a + mt$, $t \in \mathbb{Z}$. Chia t cho k, giả sử $t = kq + i$ $(q, i \in Z,\ 0 \leq i \leq k-1)$. Ta có:

$x = a + mi + mqk \equiv (a + mi) (\text{mod } km)$ nên $x \in \overline{a + im} = A_i \in \bigcup_{i=0}^{k-1} A_i$.

Ngược lại, giả sử $x \in \bigcup_{i=0}^{k-1} A_i$. Khi ấy tồn tại số nguyên i ($0 \le i \le k$ - 1) sao cho $x \in A_i$, tức là $x \equiv a + mi (\text{mod } km)$ nên $x \equiv (a + mi) (\text{mod } m)$. Do đó $x \equiv a (\text{mod } m)$, tức là $x \in A$. Vậy $A = \bigcup_{i=0}^{k-1} A_i$ và ta có điều phải chứng minh.

2.3. Tập các lớp thặng dư nguyên tố với môđun

Nhận xét

Tất cả các thặng dư của cùng một lớp thặng dư có cùng ước chung lớn nhất với môđun.

Thật vậy, giả sử $A \in \mathbb{Z}_m$ và $a, b \in A$. Khi *ấy* $a \equiv b (\text{mod } m)$ nên theo tính chất i. của đồng dư thức ta có UCLN(a, m) = UCLN (b, m). Từ đây ta có

Định nghĩa

Ước chung lớn nhất của một lớp với môđun m là ước chung lớn nhất của một thặng dư tùy ý của lớp đó với môđun m.

Với $A = \bar{a} (\text{mod } m)$, ta đặt UCLN ($A$, m) = d nếu UCLN (a, m) = d.

Khi d =1 ta nói lớp thặng dư A là *lớp nguyên tố* với môđun m.

Tập hợp các lớp $\mathbb{Z}_m$ nguyên tố với môđun được kí hiệu bởi $\mathbb{Z}^*_m$. Ta có:

$$\mathbb{Z}^*_m = \{A \in \mathbb{Z}_m \mid UCLN(A, m) = 1\} = \{A \in \mathbb{Z}_m \mid UCLN(a, m) = 1, a \in A\}.$$

Số các phần tử của tập $\mathbb{Z}^*_m$ được kí hiệu là $\varphi(m)$.

Vì $\mathbb{Z}_m = \{\bar{0}, \bar{1}, ..., \overline{m-1}\}$ nên $\mathbb{Z}^*_m = \{\bar{a} \in \mathbb{Z}_m \mid 0 \le a \le m-1, UCLN(a, m) = 1\}$.

Vậy $\varphi(m)$ chính là số các số tự nhiên không vượt quá $m-1$ và nguyên tố cùng nhau với m.

2.4. Vành các lớp thặng dư

Phép toán trong $\mathbb{Z}_m$

Trong $\mathbb{Z}_m$, ta định nghĩa phép cộng và phép nhân như sau:

Giả sử $\bar{a}, \bar{b} \in \mathbb{Z}_m$, ta đặt $\bar{a} + \bar{b} = \overline{a+b}$ và $\bar{a}.\bar{b} = \overline{ab}$.

Dễ kiểm tra được các phép toán trên là hoàn toàn xác định.

Định lý

Tập hợp $\mathbb{Z}_m$ các lớp thặng dư môđun m cùng với phép cộng và phép nhân xác định theo qui tắc trên là một vành giao hoán.

Phần tử khả nghịch

Lớp thặng dư *A* môđun *m* là phần tử khả nghịch của vành $\mathbb{Z}_m$ khi và chỉ khi *A* là lớp nguyên tố với môđun *m.*

Chứng minh

Giả sử $A = \bar{a}$ là khả nghịch, khi ấy tồn tại $B \in \mathbb{Z}_m$ sao cho $A.B = E = \bar{1}(\mathrm{mod}\, m)$, tức là $a.b \equiv 1(\mathrm{mod}\, m)$. Nếu *A* là lớp không nguyên tố với môđun *m*, tức là $(a,m) \neq 1$ thì tồn tại các số $q \neq 1$, a_1, m_1 nguyên sao cho $a = qa_1$ và $m = qm_1$. Khi ấy $ab = qa_1 b$ và $(ab,m) = q \neq 1$. Vô lý. Vậy $(A, m) = (a, m) = 1$.

Ngược lại, giả sử $(A, m) = 1$ và $A = \bar{a}$, tức là $(a, m) = 1$.

Không giảm tổng quát, có thể coi $0 < a < m - 1$.

Tập $\{0, a, 2a, ..., (m-1)a\}$ chứa phần tử ab sao cho $ab \equiv 1(\mathrm{mod}\, m)$.

Thật vậy, nếu với mọi $0 \leq b < m$ ta có $ab \not\equiv 1(\mathrm{mod}\, m)$ thì theo nguyên lý Dirichlet phải có hai phần tử ab_1 và ab_2 $(0 \leq b_1 \neq b_2 < m)$ cùng có số dư khi chia cho *m*, nghĩa là $ab_1 - ab_2 = a(b_1 - b_2) = km$. Nhưng $0 < |b_1 - b_2| < m$ nên $(a,m) \neq 1$, vô lý. Nghĩa là tồn tại $0 < b < m$ sao cho $ab \equiv 1(\mathrm{mod}\, m)$.

Đặt $B = \bar{b}$, ta có $\overline{ab} = \bar{a}.\bar{b} = \bar{1}$ hay $AB = E$, nghĩa là *A* khả nghịch.

Tính chất của phần tử khả nghịch

Giả sử A, B là những lớp thặng dư của vành $\mathbb{Z}_m$ và A khả nghịch. Khi X chạy qua tất cả các lớp thặng dư của vành $\mathbb{Z}_m$ thì $AX+B$ cũng chạy qua tất cả các phần tử của $\mathbb{Z}_m$ và AX cũng chạy qua tất cả các phần tử khả nghịch của $\mathbb{Z}_m$, tức là:

$$\mathbb{Z}_m=\left\{AX+B \middle| X\in\mathbb{Z}_m\right\} \text{ và } \mathbb{Z}_m^*=\left\{AX \middle| X\in\mathbb{Z}_m^*\right\}.$$

Kí hiệu $\varphi(m)$ là số các phần tử khả nghịch của vành $\mathbb{Z}_m$ các lớp thặng dư môđun m, hay $\varphi(m)=\text{card}(\mathbb{Z}_m^*)$.

Ta biết rằng $\mathbb{Z}_m=\left\{\bar{0},\bar{1},\bar{2},...,\overline{m-1}\right\}$, từ đó ta có

$$\mathbb{Z}_m^*=\left\{\bar{n}\in\mathbb{Z}_m \middle| 0\le n\le m-1,(n,m)=1\right\}.$$

Như vậy ta được $\varphi(m)=\text{card}(\mathbb{Z}_m^*)=\sum_{\substack{0\le n\le m-1\\(n,m)=1}}1$, nghĩa là $\varphi(m)$ là hàm số biểu thị các số tự nhiên không lớn hơn $m-1$ và nguyên tố cùng nhau với m.

Ta cũng có thể viết $\mathbb{Z}_m=\left\{\bar{1},\bar{2},...,\bar{m}\right\}$, khi ấy

$$\mathbb{Z}_m^*=\left\{\bar{n}\in\mathbb{Z}_m \middle| 1\le n\le m,(n,m)=1\right\}.$$

Như vậy ta được $\varphi(m)=\text{card}(Z_m^*)=\sum_{\substack{1\le n\le m\\(n,m)=1}}1$, nghĩa là $\varphi(m)$ là hàm số biểu thị các số tự nhiên khác không, không lớn hơn m và nguyên tố với m.

Hệ quả

$\varphi(1)=1$ và nếu p là số nguyên tố thì ta có thì ta có $\varphi(m)=p-1$.

§3. Hệ thặng dư đầy đủ - Hệ thặng dư thu gọn

3.1. Hệ thặng dư đầy đủ

Cho m là một số nguyên dương. Tập H gồm nhũng số nguyên lấy ra ở mỗi lớp thặng dư của $\mathbb{Z}_m$ một và chỉ một số được gọi là một *hệ thặng dư đầy đủ* môđun m.

Như vậy: Tập hợp H gồm những số nguyên là một hệ thặng dư đầy đủ môđun m khi và chỉ khi:

- Các phần tử của H đôi một không đồng dư với nhau theo môđun m.
- Mỗi số nguyên đều đồng dư theo môđun m với một số nào đó thuộc H.

Mỗi một số nguyên của H được gọi là một thặng dư.

Ví dụ với $m = 8$ ta có: $Z_8 = \{\bar{0}, \bar{1}, \bar{2}, \bar{3}, \bar{4}, \bar{5}, \bar{6}, \bar{7}\}$ là một hệ thặng dư đầy đủ môđun 8, nó được gọi là hệ thặng dư đầy đủ không âm nhỏ nhất. Còn hệ $\{\overline{-3}, \overline{-2}, \overline{-1}, \bar{0}, \bar{1}, \bar{2}, \bar{3}\}$ là một hệ thặng dư môđun 8, hệ này được gọi là hệ thặng dư đầy đủ giá trị tuyệt đối nhỏ nhất.

Tổng quát

+) $H = \{0, 1, ..., m - 1\}$ là một hệ thặng dư đầy đủ môđun m và nó là hệ thặng dư đầy đủ không âm nhỏ nhất.

+) Với m là một số lẻ, ta có

$$H = \left\{-\frac{m-1}{2}; -\frac{m-1}{2}+1; ...; \frac{m-1}{2}\right\}$$

là một hệ thặng dư đầy đủ môđun m được gọi là hệ thặng dư đầy đủ giá trị tuyệt đối nhỏ nhất.

+) Với m là một số chẵn, ta có

$$H = \left\{-\frac{m}{2}; -\frac{m}{2}+1; ...; \frac{m}{2}\right\} \text{ hay } H = \left\{-\frac{m}{2}+1; -\frac{m}{2}+2; ...; \frac{m}{2}\right\}$$

là hệ thặng dư đầy đủ giá trị tuyệt đối nhỏ nhất.

Tính chất 1

Mỗi hệ thặng dư đầy đủ môđum m đều gồm m phần tử.

Chứng minh

Hiển nhiên vì tập $\mathbb{Z}_m$ có m phần tử.

Tính chất 2

Mỗi hệ gồm m số nguyên đôi một không đồng dư với nhau theo môđun m đều là một hệ thặng dư đầy đủ môđun m.

Chứng minh

Giả sử $H=\{a_1, a_2,...,a_n\}$ là một hệ gồm m số nguyên đôi một không đồng dư với nhau theo môđun m. Khi ấy tập các lớp thặng dư theo môđun m $\{\overline{a}_1, \overline{a}_2,..., \overline{a}_m\}$ gồm m phần tử đôi một phân biệt và là tập con của $\square_m$. Nhưng vì $\square_m$ có m phần tử và tập con $\{\overline{a}_1, \overline{a}_2,..., \overline{a}_n\}$ cũng có m phần tử đôi một phân biệt nên ta có $\{\overline{a}_1, \overline{a}_2,..., \overline{a}_n\}=\square_m$.

Từ $\square_m=\{\overline{a}_1, \overline{a}_2,..., \overline{a}_n\}$ ta được $H=\{a_1, a_2,...,a_n\}$ là một hệ thặng dư đầy đủ môđun m.

Tính chất 3

Giả sử a là một số nguyên tố với m và b là một số nguyên tùy ý. Khi ấy xét x chạy qua một hệ thặng dư đầy đủ môđun m thì $ax\pm b$ cũng chạy qua một hệ thặng dư đầy đủ môđun m.

Chứng minh

Giả sử x chạy qua một hệ thặng dư môđun m là $\{x_1, x_2,..., x_m\}$. Ta chứng minh $\{ax_1+b, ax_2+b,..., ax_m+b\}$ cũng là một hệ thặng dư đầy đủ môđun m.

Theo Tính chất 2 ở trên ta chỉ cần chứng minh rằng với $1\le i\ne j\le m$ thì $ax_i+b\not\equiv(ax_j+b)(\mathrm{mod}\,m)$. Thật vậy, nếu $ax_i+b\equiv ax_j+b\,(\mathrm{mod}\,m)$ thì $ax_i\equiv ax_j\,(\mathrm{mod}\,m)$. Vì a nguyên tố với m nên $x_i\equiv x_j\,(\mathrm{mod}\,m)$. Vô lý vì x_i và x_j là 2 thặng dư khác nhau của một hệ thặng dư đầy đủ môđun m.

Vậy $\{ax_1+b, ax_2+b,..., ax_m+b\}$ cũng là một hệ thặng dư đầy đủ môđun m.

3.2. Hệ thặng dư thu gọn

Cho *m* là một số nguyên dương. Tập hợp *K* gồm những số nguyên được lấy ra ở mỗi lớp nguyên tố với môđun *m* một và chỉ một số được gọi là một *hệ thặng dư thu gọn* môđun *m*.

Vậy một tập hợp *K* gồm những số nguyên được gọi là một hệ thặng dư thu gọn môđun *m* nếu và chỉ nếu:

- Các phần tử thuộc *K* đôi một không đồng dư với nhau theo môđun *m*.
- Các phần tử thuộc *K* nguyên tố với môđun *m*.
- Mỗi số nguyên tùy ý nguyên tố với môđun *m* đều đồng dư với một số nào đó thuộc *K*.

Nhận xét

Mỗi hệ thặng dư đầy đủ đều chứa duy nhất một hệ thặng dư thu gọn.

Hệ thặng dư thu gọn không âm nhỏ nhất $\{r_1, r_2, ..., r_{\varphi(m)}\}$ là hệ thặng dư thu gọn gồm các phần tử $0 < r_i < m, i = 1,2,...,\varphi(m)$ nguyên tố cùng nhau với *m*.

Ta có khái niệm hệ thặng dư thu gọn môđun *m* có trị tuyệt đối nhỏ nhất.

Ví dụ

Khi *m* = 8 ta có $\{1, 3, 5, 7\}$ là một hệ thặng dư thu gọn không âm nhỏ nhất. Hệ $\{\overline{-3}, , \overline{-1}, \overline{0}, \overline{1}, \overline{3}\}$ là một hệ thặng dư thu gọn giá trị tuyệt đối nhỏ nhất.

Nếu $m = p$ là một số nguyên tố thì $\{1, 2, ..., p-1\}$ là hệ thặng dư thu gọn không âm nhỏ nhất và nếu $p > 2$ thì $\left\{-\frac{p-1}{2}, ..., -2, -1, 0, 1, 2, ..., \frac{p-1}{2}\right\}$ là hệ thặng dư thu gọn giá trị tuyệt đối nhỏ nhất.

Tính chất của hệ thặng dư thu gọn

Tính chất 1

Mỗi hệ thặng dư thu gọn môđun *m* gồm $\varphi(m)$ phần tử.

Chứng minh

Hiển nhiên vì tập hợp $\mathbb{Z}_m^*$ có $\varphi(m)$ phần tử.

Tính chất 2

Mỗi hệ gồm $\varphi(m)$ số nguyên tố với *m* và đôi một không đồng dư với nhau theo môđun *m* đều lập nên một hệ thặng dư thu gọn môđun *m*.

Chứng minh

Giả sử $K=\{a_1, a_2,...,a_{\varphi(m)}\}$ là một hệ gồm $\varphi(m)$ số nguyên nguyên tố với *m* và đôi một không đồng dư với nhau theo môđun *m*.

Vì $a_1, a_2,...,a_{\varphi(m)}$ nguyên tố với *m* nên ta có tập hợp $\{\bar{a}_1, \bar{a}_2,...,\bar{a}_{\varphi(m)}\}$ các lớp theo môđun *m* là một tập con của $\mathbb{Z}_m^*$ gồm $\varphi(m)$ phần tử, nghĩa là có số phần tử bằng số phần tử của $\mathbb{Z}_m^*$, do đó ta có $\{\bar{a}_1, \bar{a}_2,...,\bar{a}_{\varphi(m)}\} = Z_m^*$.

Từ $\mathbb{Z}_m^* = \{\bar{a}_1, \bar{a}_2,...,\bar{a}_{\varphi(m)}\}$ ta được $K=\{a_1, a_2,...,a_{\varphi(m)}\}$ là một hệ thặng dư thu gọn môđum *m*.

Tính chất 3

Giả sử *a* là một số nguyên tố với *m*. Khi ấy nếu *x* chạy qua một hệ thặng dư thu gọn môđun *m* thì *ax* cũng chạy qua một hệ thặng dư thu gọn môđun *m*.

Chứng minh

Giả sử *x* chạy qua hệ thặng dư thu gọn $\{x_1, x_2,..., x_{\varphi(m)}\}$ môđun *m*. Khi ấy $\{ax_1, ax_2,..., ax_{\varphi(m)}\}$ cũng là một hệ thặng dư thu gọn môđun *m*.

Thật vậy, $\{ax_1, ax_2,..., ax_{\varphi(m)}\}$ là một hệ gồm $\varphi(m)$ số nguyên nguyên tố với *m* vì UCLN(*a*, *m*) = 1 và UCLN(x_i, m) = 1, (*i* = 1, 2, ..., $\varphi(m)$). Theo tính chất 2 ở trên, ta chỉ cần chứng minh rằng với $i \neq j, 1 \leq i, j \leq \varphi(m)$ thì $ax_i \not\equiv ax_j \pmod m$. Giả sử ngược lại, $ax_i \equiv ax_j \pmod m$. Do UCLN$(a,m)=1$ ta

có $x_i \equiv x_j \pmod m$, $1 \le i, j \le \varphi(m)$. Điều này mâu thuẫn với giả thiết với x_i, x_j là hai thặng dư khác nhau của một hệ thặng dư thu gọn.

Vậy $\{ax_1, ax_2, ..., ax_{\varphi(m)}\}$ cũng là một hệ thặng dư thu gọn môđun *m*.

3.3. Các định lí quan trọng

Định lý Euler

Giả sử *m* là một số tự nhiên lớn hơn 1 và a là một số nguyên tố cùng nhau với *m*. Khi ấy ta có

$$a^{\varphi(m)} \equiv 1 \pmod m.$$

Chứng minh

Ta cho x chạy qua hệ thặng dư thu gọn môđun *m* không âm nhỏ nhất $\{r_1, r_2, ..., r_{\varphi(m)}\}$. Khi ấy theo tính chất 3, tập hợp $\{ar_1, ar_2, ..., ar_{\varphi(m)}\}$ cũng là một hệ thặng dư thu gọn môđun *m*.

Giả sử $s_1, s_2, ..., s_{\varphi(m)}$ là các thặng dư không âm nhỏ nhất tương ứng cùng lớp với $ar_1, ar_2, ..., ar_{\varphi(m)}$, tức là $0 < s_i < m, 1 \le i \le \varphi(m)$ và

$$ar_1 \equiv s_1 \pmod m; ar_2 \equiv s_2 \pmod m; \ldots; ar_{\varphi(m)} \equiv s_{\varphi(m)} \pmod m. \qquad (3.1)$$

Khi ấy $\{s_1, s_2, ..., s_{\varphi(m)}\}$ cũng là hệ thặng dư thu gọn môđun *m* không âm nhỏ nhất.

Vì $\{r_1, r_2, ..., r_{\varphi(m)}\}$ và $\{s_1, s_2, ..., s_{\varphi(m)}\}$ cùng là hệ thặng dư thu gọn môđun *m* không âm nhỏ nhất nên ta có $r_1 r_2 ... r_{\varphi(m)} = s_1 s_2 ... s_{\varphi(m)}$.

Nhân vế với vế $\varphi(m)$ đồng dư thức (3.1) ta được

$$a^{\varphi(m)} r_1 r_2 ... r_{\varphi(m)} \equiv s_1 s_2 ... s_{\varphi(m)} \pmod m.$$

Vì $(r_i, m) = 1$, (i = 1, 2, ..., $\varphi(m)$), nên $a^{\varphi(m)} \equiv 1 \pmod m$.

Định lý Euler được chứng minh.

Định lý Fermat

Cho p là một số nguyên tố và a là một số nguyên không chia hết cho p. Khi ấy ta có: $a^{p-1} \equiv 1 (\mathrm{mod}\ p)$.

Chứng minh

Theo giả thiết ta có $\varphi(p) = p - 1$ và $(a, p) = 1$.

Theo định lý Euler ta có: $a^{p-1} \equiv 1 (\mathrm{mod}\ p)$.

Định lý (Dạng khác của định lý Fermat)

Cho p là một số nguyên tố và a là một số tùy ý. Khi ấy ta có

$$a^p \equiv a (\mathrm{mod}\ p).$$

Chứng minh

Nếu a là một số nguyên chia hết cho p thì hiển nhiên $a^p \equiv a (\mathrm{mod}\ p)$. Nếu a không chia hết cho p thì $a^{p-1} \equiv 1 (\mathrm{mod}\ p)$. Do $a \equiv a (\mathrm{mod}\ p)$ nên nhân hai đồng dư thức ta được $a^p \equiv a (\mathrm{mod}\ p)$. Định lý được chứng minh.

§4. Phương trình đồng dư

4.1. Các khái niệm chung

Kí hiệu $\mathbb{Z}[x]$ là tập các đa thức một biến với các hệ số nguyên.

Giả sử $g(x)$ và $h(x)$ là những đa thức một biến x với các hệ số nguyên và m là một số tự nhiên lớn hơn 1.

Các phương trình chứa biến (ẩn) x dạng

$$g(x) \equiv h(x) (\mathrm{mod}\, m)$$

hay

$$f(x) \equiv (g(x) - h(x)) (\mathrm{mod}\, m) \qquad (4.1)$$

được gọi là *phương trình đồng dư* một ẩn.

Nhận xét rằng, ở đây, phương trình (4.1) chỉ là một trường hợp riêng của phương trình đồng dư nhiều ẩn $f(x_1, x_2, ..., x_n)$ với $f(x_1, x_2, ..., x_n)$ là một đa thức nhiều biến với hệ số nguyên.

Sau đây ta sẽ nghiên cứu phương trình đồng dư một ẩn.

Phương trình đồng dư tương đương

Cho $f(x) \in \mathbb{Z}[x]$. Nếu với $x = x_0 \in \mathbb{Z}$ ta có $f(x_0) \equiv 0 \pmod{m}$ thì ta nói x_0 nghiệm đúng phương trình $f(x) \equiv 0 \pmod{m}$.

Giải một phương trình đồng dư là tìm tập hợp các giá trị nghiệm đúng phương trình đồng dư đó.

Giả sử $g(x), f(x) \in \mathbb{Z}[x]$. Hai phương trình đồng dư

$$g(x) \equiv 0 (\text{mod } m_1)$$

$$f(x) \equiv 0 (\text{mod } m_2)$$

tương đương với nhau nếu như tập hợp các giá trị nghiệm đúng phương trình này bằng tập hợp các giá trị nghiệm đúng phương trình kia.

Khi ấy ta viết: $g(x) \equiv 0 (\text{mod } m_1) \Leftrightarrow f(x) \equiv 0 (\text{mod } m_2)$.

Định nghĩa

Phép biến đổi một phương trình đồng dư thành một phương trình đồng dư khác tương đương với nó được gọi là *phép biến đổi tương đương*.

Hiển nhiên hai phương trình đồng dư cùng tương đương với phương trình đồng dư thứ ba thì tương đương với nhau.

Các phép biến đổi tương đương thường gặp

a) Cộng hay trừ hai vế của một phương trình đồng dư cùng với một đa thức có hệ số là những số nguyên thì được một phương trình mới tương đương.

b) Nếu ta thêm hay bớt ở một vế của một phương trình đồng dư theo môđun m một bội của môđun m thì ta được một phương trình mới tương đương

c) Xét phương trình đồng dư

$$f(x) \equiv 0 \pmod{m}$$

với

$$f(x) = a_0x^n + a_1x^{n-1} + ... + a_n,\ a_i \in \mathbb{Z},\ i = 0, 1, \ldots, n.$$

Nếu ta nhân các hệ số của $f(x)$ với một số nguyên, nguyên tố với môđun m thì ta được một phương trình mới tương đương.

Nếu ta chia các hệ số của $f(x)$ cho cùng một ước chung nguyên tố với môđun m thì ta được một phương trình mới tương đương.

Nếu ta nhân các hệ số của $f(x)$ và môđun m với cùng một số nguyên dương thì ta được một phương trình mới tương đương.

Chia các hệ số của $f(x)$ và môđun m với cùng một ước chung dương của chúng thì được một phương trình mới tương đương với phương trình đã cho.

Bậc của phương trình đồng dư

Xét phương trình đồng dư

$$f(x) \equiv 0 \pmod m \qquad (4.2)$$

với $f(x) = a_0x^n + a_1x^{n-1} + ... + a_n,\ a_i \in \mathbb{Z},\ i = 0, 1, \ldots, n.$

Nếu $a_0 \not\equiv 0 \pmod m$ thì ta nói n là *bậc* của phương trình đồng dư (4.2).

Ví dụ

Cho phương trình: $15x^6 - 8x^4 + x^2 + 6x + 8 \equiv 0 \pmod 3$

Ta thấy $15 \equiv 0 \pmod 3$ nên bậc của phương trình không phải là bậc 6. Phương trình trên tương đương với phương trình $-8x^4 + x^2 + 2 \equiv 0 \pmod 3$. Vì $-8 \not\equiv 0 \pmod 3$ nên bậc của phương trình là $n = 4$.

Chú ý

- Trong phương trình (4.2) ta có thể giả thiết a_0 không chia hết cho m.

Thật vậy, nếu $a_0 \equiv 0 \pmod m$ thì ta có thể bỏ số hạng a_0x^n ở phương trình (4.2), ta vẫn được một phương trình tương đương với phương trình (4.2).

- Trong phương trình (4.2) ta có thể đưa các hệ số $a_0, a_1, ..., a_n$ về các số nguyên không âm nhỏ thua m.

Thật vậy, với a_0 chẳng hạn, ta chia a_0 cho m ta được: $a_0 = mq + a_0'$, $q, a_0' \in \square$, $0 \le a_0' \le m$. Khi ấy, phương trình (4.2) tương đương với phương trình $a_0' x^n + a_1 x^{n-1} + ... + a_n \equiv 0 (\mathrm{mod}\, m)$, $0 \le a_0' \le m$.

Nghiệm của phương trình đồng dư

Tập các giá trị nghiệm đúng của phương trình $f(x) \equiv 0 (\mathrm{mod}\, m)$ thường được phân chia thành những lớp theo môđun m và được gọi là những nghiệm của phương trình đó.

Định lý

Nếu $x = \alpha$ là nghiệm đúng phương trình (4.2) thì mọi số nguyên thuộc lớp thặng dư $\overline{\alpha} (\mathrm{mod}\, m)$ đều nghiệm đúng phương trình (4.2).

Chứng minh

Thật vậy, theo giả thiết ta có $f(\alpha) \equiv 0 (\mathrm{mod}\, m)$. Giả sử $\beta \in \overline{\alpha} (\mathrm{mod}\, m)$, nghĩa là $\beta \equiv \alpha (\mathrm{mod}\, m)$. Theo tính chất của đồng dư thức ta được

$f(\beta) \equiv f(\alpha) (\mathrm{mod}\, m)$. Suy ra β cũng là nghiệm của phương trình (4.2).

Định nghĩa

Khi số nguyên α nghiệm đúng phương trình (4.2) thì ta gọi lớp thặng dư $\overline{\alpha} (\mathrm{mod}\, m)$ là một nghiệm của phương trình (4.2).

Khi $\overline{\alpha} (\mathrm{mod}\, m)$ là một nghiệm của phương trình (4.2) thì ta cũng viết $x \equiv \alpha (\mathrm{mod}\, m)$ và gọi x là một nghiệm của phương trình (4.2).

Hệ quả

Số nghiệm của một phương trình đồng dư theo môđun m không vượt quá m. Do đó để giải phương trình đồng dư ta lần lượt cho x lấy các giá trị trong một hệ thặng dư đầy đủ và tìm các giá trị nghiệm đúng phương trình đó.

Ví dụ

Giải phương trình $2x^3 + 4 \equiv 0 (\text{mod } 5)$.

Cho x nhận lần lượt các giá trị hệ thặng dư đầy đủ $\square_5 = \{0,1,2,3,4\}$, ta thấy:

$$x = 2 \Rightarrow 2x^3 \equiv -4 (\text{mod } 5).$$

Vậy $x = 2$ là nghiệm duy nhất của phương trình đã cho.

Hệ phương trình đồng dư

Cho hệ phương trình

$$\begin{cases} f_1(x) \equiv 0 (\text{mod } m_1) \\ f_2(x) \equiv 0 (\text{mod } m_2) \\ \quad .... \\ f_r(x) \equiv 0 (\text{mod } m_r) \end{cases} \qquad (4.3)$$

Nếu với số nguyên $x = x_0$ ta có r đồng dư thức $f_i(x) \equiv 0 (\text{mod} m_i)$ đúng với mọi $i = 1, 2, \ldots, r$ thì ta nói x_0 nghiệm đúng hệ phương trình (4.3).

Giải một hệ phương trình đồng dư là tìm tập hợp các giá trị nghiệm đúng hệ phương trình đồng dư đó.

Hệ phương trình tương đương

Hai hệ phương trình đồng dư

$$f_i(x) \equiv 0 (\text{mod } m_i),\ i = 1, 2, \ldots, r;$$

$$g_i(x) \equiv 0 (\text{mod } n_i),\ j = 1, 2, \ldots, s$$

được gọi là *tương đương* nếu tập hợp các giá trị nghiệm đúng hệ phương trình này trùng với tập hợp các giá trị nghiệm đúng hệ phương trình kia.

Do vậy, nếu trong một hệ ta thay thế một số phương trình nào đó bằng những phương trình tương đương thì ta sẽ được một hệ mới tương đương với hệ đã cho. Do đó, việc biến đổi tương đương các hệ phương trình thường đưa về việc biến đổi tương đương từng phương trình.

Định lý

Cho *m* là một số tự nhiên có dạng phân tích tiêu chuẩn $m = p_1^{\alpha_1} p_2^{\alpha_2} ... p_k^{\alpha_k}$ và *f*(*x*) là một đa thức với hệ số nguyên. Khi ấy ta có phương trình đồng dư

$$f(x) \equiv 0 (\text{mod } m) \tag{4.4}$$

tương đương với hệ

$$f(x) \equiv 0 (\text{mod } p_i^{\alpha_i}) \frac{1}{2},\ i = 1, 2, \ldots, k. \tag{4.5}$$

Chứng minh

Thật vậy, giả sử x_0 nghiệm đúng phương trình (4.4), nghĩa là

$$f(x_0) \equiv 0 (\text{mod} m).$$

Khi đó vì *i* = 1, 2, …, *k* có $p_i^{\alpha_i}$ là ước của m nên ta có $f(x_0) \equiv 0 (\text{mod } p_i^{\alpha_i})$, nói khác đi x_0 nghiệm đúng hệ (4.5).

Ngược lại, giả sử x_1 nghiệm đúng hệ phương trình (4.5), nghĩa là ta có đồng dư thức $f(x_1) \equiv 0 (\text{mod } p_i^{\alpha_i})$, *i* =1, 2, …, *k*.

Do $m = p_1^{\alpha_1} p_2^{\alpha_2} ... p_k^{\alpha_k}$ nên ta cũng có đồng dư thức $f(x_1) \equiv 0 (\text{mod } p_i^{\alpha_i})$, tức x_1 cũng nghiệm đúng phương trình (4.4).

Nghiệm của hệ phương trình đồng dư

Cho hệ phương trình

$$\begin{cases} f_1(x) \equiv 0 (\text{mod} m_1) \\ f_2(x) \equiv 0 (\text{mod} m_2) \\ \quad .... \\ f_r(x) \equiv 0 (\text{mod} m_r) \end{cases} \tag{4.6}$$

với *m* là bội chung nhỏ nhất của $m_1, m_2, ..., m_r$.

Định lý

Nếu $x = \alpha$ nghiệm đúng hệ phương trình (4.6) thì mọi số nguyên thuộc lớp $\bar{\alpha} (\text{mod} m)$ đều nghiệm đúng hệ phương trình đó.

Chứng minh

Theo giả thiết ta có r đồng dư thức

$$f_i(x) \equiv 0 (\mathrm{mod}\, m_i),\ i = 1, 2, \ldots, r. \qquad (4.7)$$

Giả sử $\beta \in \overline{\alpha}\ (\mathrm{mod}\, m)$ nghĩa là $\beta \equiv \alpha (\mathrm{mod}\, m)$, vì vậy do m_i là ước của m, $i = 1,\ 2, \ldots, r$ nên ta có $\beta \equiv \alpha (\mathrm{mod}\ m_i)$.

Theo tính chất của đồng dư thức ta được r đồng dư thức

$$f_i(\beta) \equiv f_i(\alpha) (\mathrm{mod}\ m_i),\ i = 1, 2, \ldots, r\ . \qquad (4.8)$$

Từ đồng dư thức (4.7) và (4.8) ta có $f_i(\beta) \equiv 0 (\mathrm{mod}\ m_i)$, $i = 1, 2, \ldots, r$, nghĩa là β nghiệm đúng hệ phương trình (4.6).

Định nghĩa

Khi số nguyên α nghiệm đúng hệ phương trình (4.6) thì ta gọi lớp thặng dư $\overline{\alpha}(\mathrm{mod}\, m)$ là *nghiệm* của hệ phương trình (4.6).

4.2. Phương trình và hệ phương trình đồng dư bậc nhất một ẩn

4.2.1. Phương trình đồng dư bậc nhất một ẩn

$$ax + b \equiv c (\mathrm{mod}\ m),\quad a \neq 0 \qquad (4.8)$$

Trường hợp 1: $a = 1$

Khi $a = 1$ phương trình (4.8) trở thành

$$x + b \equiv c (\mathrm{mod}\ m) \Leftrightarrow x \equiv c (\mathrm{mod}\ m) - b \Leftrightarrow x \equiv \big(c - b\big) (\mathrm{mod}\ m)$$

$\Leftrightarrow x = c - b + mt$ với t là một số nguyên bất kỳ.

Trường hợp 2: $b = 0$

Khi $b = 0$ phương trình (4.8) trở thành

$$ax \equiv c (\mathrm{mod}\ m). \qquad (4.9)$$

Định lý

Phương trình (4.9) có nghiệm khi và chỉ khi ước chung lớn nhất d của a và m là ước của c. Khi (4.9) có nghiệm thì nó có d nghiệm.

Chứng minh

Giả sử phương trình (4.9) có nghiệm, nghĩa là có $x_0 \in \square$ sao cho $ax_0 \equiv c \pmod m$. Vì $d = (a, m)$ nên $d \mid a$ và $d \mid m$. Suy ra $d \mid ax_0$ và $d \mid m$.

Theo tính chất của đồng dư thức, d phải là một ước số của $\text{UCLN}(ax_0, m) = \text{UCLN}(c, m)$ hay d là ước của c.

Ngược lại, giả sử $(a,m)=d$ là ước của c.

Đặt $a = a = a_1 d$, $c = c_1 d$, $m = m_1 d$. Phương trình (4.9) tương đương với

$$a_1 x \equiv c_1 \pmod{m_1}, \qquad (4.10)$$

trong đó $(a_1, m) = 1$. Do $(a_1, m) = 1$ nên khi cho x chạy qua một hệ thặng dư đầy đủ môđun m_1 thì $a_1 x$ cũng chạy qua một hệ thặng dư đầy đủ môđun m_1.

Do đó tìm được duy nhất một giá trị x_0 sao cho $a_1 x_0$ cùng lớp với c_1, nghĩa là $a_1 x_0 \equiv c_1 \pmod{m_1}$.

Vậy phương trình (4.10) có nghiệm duy nhất là lớp $\overline{x}_0 \pmod{m_1}$. Vì phương trình (4.10) tương đương với phương trình (4.9) cho nên lớp $\overline{x}_0 \pmod{m_1}$ cũng là tập hợp các giá trị nghiệm đúng phương trình (4.9). Theo tính chất 2 §2 của đồng dư thức, lớp $\overline{x}_0$ là hợp của d lớp thặng dư môđun m, đó chính là d nghiệm của phương trình (4.9):

$$\overline{x_0} \pmod m,\ \overline{x_0 + m_1} \pmod m, \ldots,\ \overline{x_0 + (d-1)m_1} \pmod m.$$

Các phương pháp tìm nghiệm của $ax \equiv c \pmod m$

Theo Định lý trên, ta chỉ cần tìm nghiệm của phương trình (4.9) với điều kiện $(a, m) = 1$ và $1 < a < m$.

<u>Phương pháp 1:</u> Xác định nghiệm bằng cách chia cả hai vế cho *a*

- Nếu a là một ước của c thì nghiệm của phương trình là: $x \equiv \frac{c}{a} \pmod m$.

• Nếu a không phải là một ước số của c thì do $(a, m) = 1$ nên tồn tại số nguyên k $(1 \le k \le a-1)$ để $c + km$ chia hết cho a. Thật vậy, giả sử với mọi k $(1 \le k \le a-1)$, $c+km$ chia hết cho a. Khi ấy theo nguyên lí Dirichlet phải tồn tại hai số $0 \le k_1 \ne k_2 \le a-1$ sao cho $c + k_1 m$ và $c + k_2 m$ chia cho a có cùng số dư, tức là $(k_1 - k_2)m = (c + k_1 m) - (c + k_2 m)$ chia hết cho a. Nhưng $(a, m) = 1$ nên $(k_1 - k_2)$ chia hết cho a. Vô lí vì $0 < |k_1 - k_2| < a$.

Như vậy, phải tồn tại số nguyên k $(1 \le k \le a-1)$ để $c+km$ chia hết cho a. Khi ấy phương trình (4.9) tương đương với phương trình $ax \equiv c+km (\text{mod} m)$ nên nó có nghiệm là $x \equiv \dfrac{c+km}{a} (\text{mod } m)$.

Ví dụ

Giải phương trình $5x \equiv 2 (\text{mod } 7)$

Vì $\text{UCLN}(5,2) = 1$ nên tồn tại số $k = 4$ sao cho $2 + k.7$ chia hết cho 5. Khi ấy $5x \equiv 2 + 4.7 (\text{mod } 7)$ ta được nghiệm là $x \equiv \dfrac{30}{5} \equiv 6 (\text{mod } 7)$ hay $x = 6 + 7k$.

Chú ý

Cách xác định nghiệm này là đơn giản nhưng chỉ dùng được trong trường hợp a là một số nhỏ hoặc trường hợp dễ thấy ngay số k.

<u>Phương pháp 2:</u> Xác định nghiệm bằng cách vận dụng định lí Euler

Vì $(a, m) = 1$ cho nên theo Định lý Euler ta có $a^{\varphi(m)} \equiv 1 (\text{mod } m)$.

Từ đây suy ra $a^{\varphi(m)} b \equiv b (\text{mod} m)$ hay $a\left[ba^{\varphi(m)-1}\right] \equiv b (\text{mod } m)$.

Do $(a, m) = 1$ nên $x \equiv ba^{\varphi(m)-1} (\text{mod } m)$ là nghiệm của phương trình.

Trường hợp 3 $b \ne 0$

Khi $b \ne 0$ thì phương trình (4.8) trở thành

$$(4.8) \Leftrightarrow ax \equiv c (\text{mod } m) - b \Leftrightarrow ax \equiv (c-b) (\text{mod } m).$$

Trở về phương trình dạng (4.9) xét trong Trường hợp 2.

Quan hệ giữa phương trình đồng dư bậc nhất và phương trình Diophantus bậc nhất hai ẩn *ax + by = c*

Coi $b > 0$. Nếu phương trình Diophantus $ax + by = c$ có một nghiệm nguyên (x_0, y_0), nghĩa là ta có đẳng thức $a x_0 + b y_0 = c$ thì suy ra:

$$a x_0 \equiv c \pmod{b} \text{ và } y_0 = \frac{c - ax_0}{b}.$$

Đảo lại, nếu có số nguyên x_0 sao cho có đồng dư thức $a x_0 \equiv c \pmod{b}$ thì $c - ax_0 \equiv 0 \pmod{b}$, nghĩa là có số nguyên y_0 để $c - ax_0 = by_0$. Từ đó suy ra $a x_0 + b y_0 = c$, tức phương trình đã cho có nghiệm (x_0, y_0).

Vậy điều kiện phương trình Diophantus $ax + by = c$ có nghiệm nguyên tương đương với điều kiện phương trình đồng dư $a x \equiv c \pmod{b}$ có nghiệm, tức là ước chung lớn nhất của a và b là một ước của c. Giải phương trình Diophantus $ax+by=c$ được đưa về giải phương trình đồng dư $a x \equiv c \pmod{b}$.

Ví dụ

Tìm nghiệm nguyên của phương trình $1998x + 2003y = 1945$.

Ta xét phương trình đồng dư

$$1998x \equiv 1945 \pmod{2003}$$

$$\Leftrightarrow -5x \equiv 1945 \pmod{2003} \Leftrightarrow -5x \equiv 11960 \pmod{2003}$$

Do (5, 2003) = 1 nên $x = -389 \pmod{2003} \Leftrightarrow x = -389 + 2003t,\ t \in \mathbb{Z}$.

Khi ấy ta được:

$$y = \frac{1945 - 1998x}{2003} = \frac{1945 - 1998(-389 + 2003t)}{2003} = 389 - 1998t.$$

Vậy nghiệm tổng quát của phương trình là

$$\begin{cases} x = -389 + 2003t \\ y = 389 - 1998t \end{cases}, \quad (t = 0, \pm 1, \ldots)$$

4.2.2. Hệ phương trình đồng dư bậc nhất một ẩn

Xét hệ phương trình đồng dư bậc nhất một ẩn có dạng

$$\begin{cases} x \equiv b_1 (\text{mod } m_1) \\ x \equiv b_2 (\text{mod } m_2) \\ .... \\ x \equiv b_k (\text{mod } m_k) \end{cases} \qquad (4.11)$$

với $m_1, m_2, ..., m_k$ là những số nguyên lớn hơn 1 và $b_1, b_2, ..., b_k$ là những số nguyên tùy ý.

Định lý

Nếu hệ phương trình (4.11) có nghiệm thì nó có nghiệm duy nhất.

Chứng minh

Giả sử α và β nghiệm đúng hệ phương trình (4.11), ta có:

$\alpha \equiv b_i (\text{mod } m_i)$ và $\beta \equiv b_i (\text{mod } m_i)$ với mọi $i = 1, 2, \ldots, k$.

Suy ra $\alpha \equiv \beta (\text{mod } m_i)$ với mọi $i = 1, 2, \ldots, k$.

Do đó $\alpha \equiv \beta (\text{mod } m)$, trong đó m = BCNN$(m_1, m_2, ..., m_k)$, tức là các nghiệm $x \equiv \alpha (\text{mod } m)$ và $x \equiv \beta (\text{mod } m)$ của hệ phương trình (4.11) là trùng nhau.

Định lý Trung Hoa về thặng dư

Nếu các $m_1, m_2, ..., m_k$ đôi một nguyên tố cùng nhau thì hệ phương trình (4.11) có nghiệm.

Chứng minh

Theo giả thiết $m_1, m_2, ..., m_k$ đôi một nguyên tố cùng nhau nên bội chung nhỏ nhất của chúng là $m = m_1 m_2 ... m_k$. Đặt $m = m_i M_i$ với i =1,2,…,k. Khi đó ta có $(m_i, M_i) = 1$ và $M_i \equiv 0 (\text{mod } m_j)$ nếu $i \neq j$.

Vì ước chung nhỏ nhất $(m_i, M_i) = 1$ nên tồn tại số nguyên M_i' sao cho $M_i M_i' \equiv 1 (\text{mod } m_i)$, $i = 1, 2, \ldots, k$.

Đặt $x_0 = M_1 M_1' b_1 + M_2 M_2' b_2 + ... + M_k M_k' b_k$.

Vì $M_j \equiv 0 (\text{mod } m_i)$ và $M_i M_i' \equiv 1 (\text{mod } m_i)$ với mọi $i \neq j$ nên

$$x_0 \equiv b_i \pmod{m_i},\ i = 1, 2, \ldots, k.$$

Vậy x_0 thỏa mãn hệ (4.11), hay (4.11) có nghiệm là $x \equiv x_0 \pmod{m}$.

Chú ý

*) Trong trường hợp tổng quát, chúng ta có thể chứng minh được rằng: Điều kiện cần và đủ để hệ phương trình (4.11) có nghiệm là ước chung lớn nhất (m_i, m_j) chia hết $b_i - b_j$ với $i \neq j\ (1 \leq i, j \leq k)$.

*) Giả sử $m = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_k^{\alpha_k}$ là phân tích tiêu chuẩn của m. Khi ấy phương trình đồng dư $f(x) \equiv 0 (\text{mod } m)$ tương đương với hệ phương trình đồng dư $f(x) \equiv 0 \pmod{p_i^{\alpha_i}}$, $i = 1, 2, \ldots, k$.

Từ đó suy ra rằng nếu $x \equiv b_1 \pmod{p_i^{\alpha_i}}$ là một nghiệm của phương trình $f(x) \equiv 0 (\text{mod } p_i^{\alpha_i})$, $i = 1, 2, \ldots, k$. thì nghiệm của hệ phương trình đồng dư

$$\begin{cases} x \equiv b_1 \pmod{p_1^{\alpha_1}} \\ x \equiv b_2 (\text{mod } p_2^{\alpha_2}) \\ .... \\ x \equiv b_k (\text{mod } p_k^{\alpha_k}) \end{cases}$$

cho ta nghiệm của phương trình $f(x) \equiv 0 (\text{mod } m)$.

Vậy trong trường hợp tổng quát giải một phương trình đồng dư dẫn đến giải hệ (4.11). Với các môđun $m_1, m_2, ..., m_k$ đôi một nguyên tố cùng nhau.

Thực hành giải hệ phương trình

Trường hợp hệ hai phương trình

$$\begin{cases} x \equiv b_1 \pmod{m_1} \\ x \equiv b_2 \pmod{m_2} \end{cases}$$

Với giả thiết $d = (m_1, m_2)$ chia hết cho $b_1 - b_2$. Trước tiên ta nhận xét rằng, mọi số $x = b_1 + m_1 t$, $t \in \mathbb{Z}$ là nghiệm của phương trình thứ nhất. Sau đó

ta tìm cách xác định t sao cho x nghiệm đúng phương trình thứ hai, nghĩa là hệ hai phương trình trên tương đương với hệ phương trình

$$\begin{cases} x = b_1 + m_1 t \\ b_1 + m_1 t \equiv b_2 \pmod{m_2} \end{cases}$$

Vì giả thiết $d = (m_1, m_2)$ là ước $b_1 - b_2$ nên phương trình

$$b_1 + m_1 t \equiv b_2 \pmod{m_2}$$

tương đương với phương trình:

$$\frac{m_1}{d} t \equiv \frac{b_2 - b_1}{d} \left(\mod \frac{m_2}{d} \right).$$

Nhưng $\left(\frac{m_1}{d}, \frac{m_2}{d} \right) = 1$ nên phương trình đồng dư này cho ta nghiệm

$t \equiv t_0 \left(\mod \frac{m_2}{d} \right)$, là tập hợp tất cả các số nguyên $t = t_0 + \frac{m_2}{d} u$, $u \in \mathbb{Z}$.

Thay biểu thức của t vào biểu thức tính x ta được tập hợp các giá trị của x nghiệm đúng cả hai phương trình đồng dư đang xét là:

$$x = b_1 + m_1 \left(t_0 + \frac{m_2}{d} u \right) = b_1 + m_1 t_0 + \frac{m_1 m_2}{d} u,$$

hay $x = x_0 + mu$ với $x_0 = b_1 + m_1 t_0$, $m = \text{BCNN}(m_1, m_2)$.

Vậy $x \equiv x_0 (\mod m)$ là nghiệm của hệ hai phương trình đồng dư đang xét.

b. Trường hợp hệ gồm *n* phương trình

Đầu tiên giải hệ hai phương trình nào đó của hệ đã cho, rồi thay trong hệ hai phương trình đã giải bằng nghiệm tìm thấy, ta sẽ được một hệ gồm $n - 1$ phương trình tương đương với với hệ đã cho. Tiếp tục như vậy sau $n - 1$ bước ta sẽ được nghiệm cần tìm.

Ví dụ

Giải hệ phương trình $\begin{cases} x \equiv 26 \pmod{36} \\ x \equiv 62 \pmod{60} \\ x \equiv 92 \pmod{150} \\ x \equiv 11 \pmod{231} \end{cases}$

Hệ hai phương trình

$$\begin{cases} x \equiv 26 \pmod{36} \\ x \equiv 62 \pmod{60} \end{cases} \Leftrightarrow \begin{cases} x = 26 + 36t \\ 26 + 36t \equiv 62 \pmod{60} \end{cases}, t \in \mathbb{Z}.$$

Phương trình:

$26 + 36t \equiv 62 \pmod{60} \Leftrightarrow 36t \equiv 36 \pmod{60} \Leftrightarrow t \equiv 1 \pmod{5}$.

Vậy nghiệm của hệ là $x \equiv 26 + 36.1 \pmod{180}$ hay $x \equiv 62 \pmod{180}$.

Do đó hệ phương trình đã cho tương đương với hệ

$$\begin{cases} x \equiv 62 \pmod{180} \\ x \equiv 92 \pmod{150} \\ x \equiv 11 \pmod{231} \end{cases}$$

Ta tiếp tục giải hệ phương trình

$$\begin{cases} x \equiv 62 \pmod{180} \\ x \equiv 92 \pmod{150} \end{cases} \Leftrightarrow \begin{cases} x = 62 + 180t \\ 62 + 180t \equiv 92 \pmod{150} \end{cases}, t \in \mathbb{Z}.$$

Ta có: $62 + 180t \equiv 92 \pmod{150} \Leftrightarrow 180t \equiv 30 \pmod{150}$.

$\Leftrightarrow 6t \equiv 1 \pmod{5} \Leftrightarrow t \equiv 1 \pmod{5}$.

Vậy nghiệm của hệ là:

$x \equiv 62 + 180.(1+) \pmod{900} \Leftrightarrow x \equiv 242 \pmod{900}$.

Hệ đã cho tương đương với $\begin{cases} x \equiv 242 \pmod{900} \\ x \equiv 11 \pmod{231} \end{cases}$

Hệ này có nghiệm $x \equiv 242 \pmod{69300}$, và đây cũng là nghiệm của hệ đã cho cần tìm.

Ví dụ

Tìm các số nguyên chia hết cho 5 và khi lần lượt chia cho 2; 3; 4 đều có số dư là 1.

Giải

Gọi x là số cần tìm. Theo giả thiết ta có hệ phương trình:

$$\begin{cases} x \equiv 0(\text{mod } 5) \\ x \equiv 1(\text{mod } 2) \\ x \equiv 1(\text{mod } 3) \\ x \equiv 1(\text{mod } 4) \end{cases} \Leftrightarrow \begin{cases} x \equiv 0(\text{mod } 5) \\ x \equiv 1(\text{mod } 12) \end{cases}$$

Suy ra: $1 + 12t \equiv 0(\text{mod } 5) \Leftrightarrow 2t \equiv -1 \equiv 4(\text{mod } 5) \Rightarrow t \equiv 2(\text{mod}5)$.

Vậy $t = 2 + 5k,\ k \in Z$ và do đó: $x = 1 + 12t = 1 + (2 + 5k) = 25 + 60k$.

4.3. Phương trình đồng dư bậc cao theo môđun nguyên tố

4.3.1. Nhận xét

a) Giả sử $f(x) = a_0x^n + a_1x^{n-1} + ... + a_n$ là một đa thức với hệ số nguyên và m là số tự nhiên có dạng $m = p_1^{\alpha_1} p_2^{\alpha_2} ... p_k^{\alpha_k}$. Khi đó phương trình đồng dư $f(x) \equiv 0(\text{mod } m)$ tương đương với hệ $f(x) \equiv 0(\text{mod } p_i^{\alpha_i})$, i = 1, 2, …, k. Vì vậy giải phương trình $f(x) \equiv 0(\text{mod } m)$ được đưa về giải phương trình dạng $f(x) \equiv 0(\text{mod } p^{\alpha})$, với p là số nguyên tố, và α là số tự nhiên khác không.

Nếu một trong các phương trình của hệ $f(x) \equiv 0(\text{mod } p_i^{\alpha_i})$, i = 1, 2, …, k không có nghiệm thì phương trình $f(x) \equiv 0(\text{mod } m)$ cũng không có nghiệm.

Còn nếu phương trình $f(x) \equiv 0(\text{mod } p^{\alpha})$ có s_i nghiệm (i = 1, 2, …, k) thì hệ: $f(x) \equiv 0(\text{mod } p_i^{\alpha_i})$, i =1, 2,…, và do đó cả phương trình $f(x) \equiv 0(\text{mod } m)$ có $s = s_1s_2...s_k$ nghiệm.

b) Nếu số nguyên x_0 nghiệm đúng phương trình $f(x) \equiv 0(\text{mod } p^{\alpha})$, a >1 (1) thì x_0 cũng là nghiệm đúng của phương trình $f(x) \equiv 0(\text{mod } p^{\beta})$, $\beta = 1, 2, ..., \alpha - 1$.

Từ đó suy ra rằng khi giải phương trình (1) ta chỉ cần tìm các nghiệm của nó trong các lớp là nghiệm của phương trình $f(x) \equiv 0 (\text{mod } p^{\alpha-1})$.

Đối với phương trình mới này ta lại áp dụng kết quả đó để đưa về phương trình với môđun $p^{\alpha-2}$ và cứ thế tiếp tục lên đối với phương trình $f(x) \equiv 0 (\text{mod } p)$.

c) Giả sử $x = x_0 (\text{mod } p)$ là một nghiệm của phương trình $f(x) \equiv 0 (\text{mod } p)$ và đạo hàm $f'(x_0)$ không chia hết cho p khi ấy trong lớp $\overline{x}_0 (\text{mod } p)$ gồm $p^{\alpha-1}$ lớp theo môđun p^{α} có đúng một lớp là nghiệm của phương trình $f(x) \equiv 0 (\text{mod } p^{\alpha})$.

Qua những nhận xét ở trên, ta thấy rằng vấn đề cơ bản trong việc giải phương trình đồng dư còn lại là giải phương trình đồng dư theo môđun nguyên tố p: $f(x) = a_0 x^n + a_1 x^{n-1} + ... + a_n \equiv 0 (\text{mod } p)$ với $a \not\equiv 0 (\text{mod } p)$.

4.3.2. Phương trình bậc cao theo môđun nguyên tố

Định lý

Phương trình đồng dư theo môđun nguyên tố

$$f(x) = a_0 x^n + a_1 x^{n-1} + ... + a_n \equiv 0 (\text{mod } p) \qquad (4.13)$$

với $n > 1$, $a_0 \not\equiv 0 (\text{mod } p)$, hoặc nghiệm đúng với mọi số nguyên hoặc tương đương với một phương trình có bậc nhỏ hơn p.

Chứng minh

Thực hiện phép chia $f(x)$ cho $x^p - x$ ta được $f(x) = (x^p - x)g(x) + r(x)$, trong đó $g(x)$, $r(x)$ là những đa thức với hệ số nguyên, $r(x)$ hoặc bằng không hoặc có bậc nhỏ hơn p. Phương trình (4.13) trở thành

$$(x^p - x)g(x) + r(x) \equiv 0 (\text{mod } p).$$

Với mọi $x \in \mathbb{Z}$ ta đều có $x^p - x \equiv 0 (\text{mod } p)$ nên phương trình (4.13) tương đương với phương trình $r(x) \equiv 0 (\text{mod } p)$, ở đó hoặc $r(x) = 0$ hoặc có bậc nhỏ hơn p. Đó là điều cần chứng minh.

Chú ý

Theo định lý trên, trong phương trình (4.13) ta có thể giả thiết $n < p$. Hơn nữa ta còn có thể giả thiết $a_0 = 1$. Thật vậy, vì UCLN(a_0, p)=1 nên tồn tại số nguyên a sao cho $a_0 a \equiv 1 (\text{mod } p)$ và UCLN(a, p) = 1. Do đó khi nhân hai vế của phương trình (4.13) với a ta được một phương trình tương đương với phương trình (4.13) mà hệ số x^n bằng 1.

Định lý

Phương trình đồng dư theo môđun nguyên tố

$$f(x) = a_0 x^n + a_1 x^{n-1} + ... + a_n \equiv 0 (\text{mod } p) \qquad (4.14)$$

với $n > 1$, $a_0 \not\equiv 0 (\text{mod } p)$, có không quá n nghiệm.

Chứng minh

Giả sử ngược lại rằng phương trình (4.14) có ít nhất $n+1$ nghiệm khác nhau là $x \equiv x_0, x_1, ..., x_n (\text{mod } p)$.

Chia đa thức $f(x)$ cho đa thức $x - x_1$ được $f(x) = (x - x_1) f_1(x) + r_1$, trong đó là đa thức bậc $n-1$ với hệ số nguyên, với hệ số của x^{n-1} là a_0 và $r_1 = f(x_1) \equiv 0 (\text{mod } p)$. Do đó phương trình (4.14) tương đương với

$$(x - x_1) f_1(x) \equiv 0 (\text{mod } p). \qquad (4.15)$$

Từ giả thiết x_2 là nghiệm đúng phương trình (4.14) ta có đồng dư thức

$$(x_2 - x_1) f_1(x_2) \equiv 0 (\text{mod } p).$$

Nhưng $x_2 - x_1 \not\equiv 0 (\text{mod } p)$ và p là số nguyên tố nên từ đồng dư thức trên ta suy ra $f_1(x_2) \equiv 0 (\text{mod } p)$. Chia đa thức $f_1(x)$ cho đa thức $x - x_2$, giả sử ta được $f_1(x) = (x - x_2) f_2(x) + r_2$, trong đó $f_2(x)$ là đa thức bậc $n - 2$ với

hệ số nguyên, với hệ số của x^{n-2} là a_0 và $r_2 = f_1(x_2) \equiv 0(\mathrm{mod}\, p)$. Từ đây vì $r_2 \equiv 0(\mathrm{mod}\, p)$ phương trình (4.14) và do đó cả phương trình (4.13) cũng tương đương với phương trình:

$$(x - x_1)(x - x_2) f_2(x) \equiv 0(\mathrm{mod}\, p).$$

Cứ tiếp tục quá trình trên như vậy sau *n* bước ta được phương trình

$$(x - x_1)(x - x_2)\ldots(x - x_n) f_n(x) \equiv 0(\mathrm{mod}\, p)$$

tương đương với phương trình (4.14). Nhưng $f_n(x)$ là đa thức bậc không mà hệ số của số hạng bậc cao nhất là a_0 nên $f_n(x) = a_0$ và vì vậy phương trình (4.14) tương đương với phương trình

$$a_0(x - x_1)(x - x_2)\ldots(x - x_n) \equiv 0(\mathrm{mod}\, p). \qquad (4.16)$$

Theo giả thiết $x \equiv x_0(\mathrm{mod}\ p)$ là nghiệm của phương trình nên nó cũng là nghiệm của phương trình (4.16), nghĩa là ta có đồng dư thức:

$$a_0 (x - x_1)(x - x_2)\ldots(x - x_n) \equiv 0(\mathrm{mod}\, p).$$

Từ đồng dư thức này với $x_0 - x_i \not\equiv 0(\mathrm{mod}\, p)$, $\forall i = 1, 2, \ldots, n$ và p là số nguyên tố suy ra $a_0 \equiv 0(\mathrm{mod}\, p)$, điều này mâu thuẫn với giả thiết.

Hệ quả

Nếu phương trình

$$f(x) = a_0 x^n + a_1 x^{n-1} + \ldots + a_n \equiv 0(\mathrm{mod}\, p) \qquad (4.17)$$

với $n < p$ và p là một số nguyên tố có quá n nghiệm thì các hệ số của nó đều là bội của p.

Chứng minh

Thật vậy, từ giả thiết lặp lại cách chứng minh định lý trên ta có

$a_0 \equiv 0(\mathrm{mod}\, p)$ nên phương trình tương đương với

$$a_1 x^{n-1} + a_1 x^{n-1} + \ldots + a_n \equiv 0(\mathrm{mod}\, p).$$

Phương trình này có nhiều hơn n – 1 nghiệm, lại bằng cách lập luận tương tự như trên ta sẽ có $a_1 \equiv 0 (\mathrm{mod}\, p)$.

Cứ tiếp tục như vậy, cuối cùng ta được tất cả các hệ số $a_0, a_1, ..., a_n$ đều là bội của p.

Định lí Wilson

Với số nguyên tố p, ta có $(p-1)! + 1 \equiv 0 (\mathrm{mod}\ p)$.

Chứng minh

Định lí hiển nhiên đúng với $p = 2$, vì vậy để chứng minh ta giả thiết $p > 2$.

Xét phương trình đồng dư

$$(x-1)(x-2)...(x-p+1) - (x^{p-1} - 1) \equiv 0 (\mathrm{mod}\ p).$$

Phương trình có không ít hơn p – 1 nghiệm đôi một phân biệt và vế trái của nó là một đa thức bậc nhỏ hơn p – 1 vì vậy theo hệ quả trên thì tất cả các hệ số của đa thức đều là bội của p và riêng hệ số tự do cũng là bội của p. Hệ số tự do đó là: $(-1)(-2)...(-(p-1)) + 1 = (p-1)! + 1$.

Vì vậy ta có $(p-1)! + 1 \equiv 0 (\mathrm{mod}\ p)$.

Chú ý

Định lý Wilson cho ta điều kiện cần để một số tự nhiên $p > 1$ là một số nguyên tố, song điều kiện đó cũng là điều kiện đủ.

Thật vậy, nếu $p = p_1 q$, $1 < q < p$ thì $(p-1)! \equiv 0 (\mathrm{mod}\, q)$ bởi vậy $(p-1)! + 1 \not\equiv 0 (\mathrm{mod}\, q)$ nhưng vì $p \equiv 0 (\mathrm{mod}\ q)$ nên từ $(p-1)! + 1 \not\equiv 0 (\mathrm{mod}\, q)$ cũng có $(p-1)! + 1 \not\equiv 0 (\mathrm{mod}\, p)$.

Hay nếu $(p-1)! + 1 \not\equiv 0 (\mathrm{mod}\, p)$ thì số tự nhiên $p > 1$ là một số nguyên tố.

Chương 2
ỨNG DỤNG CỦA LÝ THUYẾT ĐỒNG DƯ TRONG MÃ SỬA SAI

§1. Khái niệm mã

Phần lớn các sản phẩm mà ta mua trong siêu thị đều có **mã vạch** (barcode), giống như Hình 1.1 dưới đây. Các vạch này được đọc tại các bàn thu ngân bởi hệ thống quét laze để chuyển đổi những vạch mầu đen và trắng với độ dày đậm khác nhau thành những con số được in bên dưới. Mỗi sản phẩm khi ấy được xác định bởi một chuỗi những con số, được gọi là **từ mã** (codeword).

Hình 1.1. Mã vạch

Hình 1.2. ISBN

Ở bìa sau hầu hết các quyển sách ta cũng tìm thấy các mã vạch khác nhau, thí dụ như trong Hình 1.2. Con số phía trên được gọi là **số sách tiêu chuẩn quốc tế** (International Standard Book Number, viết tắt là ISBN), và mỗi nhà xuất bản sách đều có thể được đồng nhất với một mã số theo cách này.

Khi xử lý hoặc truyền thông tin dưới dạng từ mã, các lỗi có thể xảy ra do sự cố điện, sự can thiệp từ bên ngoài như sét, bức xạ, lỗi do con người hoặc những lỗi kỹ thuật khác. Vì các nguyên nhân đó, một số chữ số trong từ mã có thể bị thay đổi, do đó cần phải tìm cách chính xác hóa lại hoặc ít nhất là phát hiện ra lỗi. Cần một số **chữ số kiểm tra** (check digits), thí dụ, chữ số 8 và chữ số 3 ở ngoài cùng bên phải tương ứng trong Hình 1.1 và Hình 1.2 để

nhìn vào đó chúng ta có thể biết chắc chắn sản phẩm ta cần mua hoặc cửa hàng gửi đến đúng quyển sách ta đã đặt. Các chữ số này được chọn bằng cách tận dụng một số tính chất cơ bản của các số trong dãy số.

Chúng ta đã quen thuộc với ngôn ngữ nói hoặc ngôn ngữ viết, trong đó chứa rất nhiều cấu trúc câu giúp chúng ta đoán được chính xác ý nghĩa của câu nói (câu viết), mặc dù câu văn chứa lỗi chính tả hoặc những lỗi khác. Thí dụ, nếu ta nhận được tin nhắn: "Meat me at fore p.n. tomorrow", ta vẫn có thể hiểu được đúng nghĩa của tin nhắn này. Việc phát hiện và sửa lỗi trong truyền tin cũng tương tự như vậy.

Thí dụ mã đơn giản

Để minh họa có thể sử dụng các con số để tạo ra một cấu trúc chặt chẽ như vốn có trong ngôn ngữ, ta xem xét ví dụ sau. Giả sử ta và một người bạn trước khi gửi đi tin nhắn đã thỏa thuận gán nhãn cho chín tin nhắn khác nhau như: "meet me at four p.m. tomorrow" (bản sửa lại đúng của tin nhắn trên), hoặc "See you at dinner tonight" bởi các số 1, 2, …, 9 tương ứng. Tiếc là ta không thể gửi đi một trong những số nguyên đó bởi vì những vấn đề kỹ thuật gây ra tới ±2 lỗi trong lúc truyền tin. Do đó, ví dụ, nếu ta nhận được số 3 thì ta không thể biết người bạn đã gửi tin nhắn nào. Có khả năng là 1 (với lỗi là 2) và 4 (với lỗi là – 1). Tuy nhiên, ta có thể bất chợt nảy ra ý tưởng: nhân số của tin nhắn với 5 và gửi đi kết quả. Ví dụ: Nếu tin nhắn ta muốn gửi cho một người bạn có nhãn là 4 thì ta gửi 4x5 = 20. Nếu lỗi truyền đi lớn nhất vẫn là ±2, thì tin nhắn luôn có thể được hiểu đúng hoặc được **giải mã** (decode) như sau. Giả sử người bạn nhận được số 22 thì anh ta suy luận đúng rằng ta đã gửi 20 với tin nhắn truyền đi sai là + 2. Vì vậy tin nhắn chỉ có thể là 20:5 = 4. Tương tự, nếu số nhận được là 38 thì ta khẳng định rằng người bạn chỉ có thể đã gửi 40 với lỗi là – 2. Vì thế 40:5=8 là nhãn của tin nhắn. Ta có thể nhận thấy rằng mọi tin nhắn là duy nhất (mỗi tin nhắn được giải mã thành một số

tương ứng với nó) và được giải mã (sửa lỗi) theo quy tắc: làm tròn lên (38 thành 40) hoặc làm tròn xuống (22 thành 20) số nhận được để được số gần nhất là bội của 5, sau đó chia số đã làm tròn cho 5 để được nhãn của tin nhắn.

Sử dụng mã để truyền và sử lý thông tin một cách chính xác tối đa là một phần thiết yếu của cuộc sống hiện đại. Chẳng hạn, ngoài mã vạch trên các sản phẩm, mã PINs (Personal Identification numbers) được sử dụng trên các thẻ lĩnh tiền tự động (cashcard); các hộ chiếu trong khối EU mang các số nhận dạng để chống giả mạo; các mã sửa sai (error-correcting codes) được sử dụng trong truyền dữ liệu từ các mạng toàn cầu nhằm bù lại những khoảng cách lớn hoặc khả năng giới hạn của các máy truyền tin. Cuối cùng nhưng không kém quan trọng, mọi đĩa compact (CD) mang dòng chữ "DIGITAL AUDIO" (âm thanh số). CD được đưa vào sử dụng năm 1982 và đã được sử dụng để tái tạo âm thanh và lưu trữ thông tin dưới dạng số. Những âm thanh này đầu tiên được phân tích thành nhiều thành phần rất mảnh và được chuyển thành các số nhị phân. Để nghe được bản nhạc, các bit được đọc trên đĩa CD bởi tia laze, và mỗi giây có 1.460.000 bit của thông tin âm thanh được xử lý. Độ dài của mỗi đoạn ghi chứa khoảng 20 tỉ bit và thậm chí với phương pháp sản xuất cẩn thận nhất, những sai sót vẫn có trên các đĩa CD. Lý do tại sao những sai sót này không ảnh hưởng đến nhạc, mà những âm thanh này còn rất chính xác và không có tiếng "click ", " hiss" và những tiếng ồn không mong muốn khác, đó là do khoảng 2/3 thông tin chứa trên đĩa CD là không dành cho thông tin âm thanh. Những thông tin thêm này được sử dụng để xử lý âm nhạc trước khi nó truyền tới tai người nghe nhằm đạt được những âm thanh cuối cùng thực sự hoàn hảo. Trong thực tế nhóm dữ liệu bao gồm 588 bit được sử dụng, trong đó 192 bit là chứa thông tin âm thanh và không nhỏ hơn 64 bit là những bit kiển tra và sửa lỗi.

Trong chương này chúng tôi trình bày một số nội dung cơ bản của mã sửa sai theo tài liệu [6], có tham khảo thêm các tài liệu [1] và [7], dựa trên ý tưởng sử dụng số học đồng dư và các trường hữu hạn đã được trình bày trong chương I.

§ 2. Những ví dụ về mã

2.1. Mã lặp

Ví dụ 2.1 (Bốn lệnh)

Giả sử ta muốn dùng một điều khiển từ xa để gửi 4 lệnh khác nhau cho máy video (video cassette recoder, VCR). Những lệnh này có thể được biểu thị bởi các **từ mã nhị phân** (binary codewords) như sau:

Lệnh	**Dừng** **Stop**	**Chơi** **play**	**Tua đi** **Fast forward (FF)**	**Tua lại** **Rewind (REW)**
Từ mã	00	01	10	11

Tuy nhiên, thậm chí một lỗi đơn giản xảy ra trong truyền tin (thí dụ 0 bị thay bởi 1 hoặc 1 bị thay bởi 0), thì lệnh sai sẽ được thực hiện bởi vì VCR không có cách để nhận biết lỗi đã xảy ra. Ví dụ nếu ta gửi 10 nhưng bị truyền sai thành 00 thì VCR dừng lại thay vì tua đi.

Trong ngôn ngữ hàng ngày nếu ta không hiểu ai đó nói, ta thường đề nghị họ nhắc lại. Bởi vậy một cách tự nhiên để sửa các lỗi là nhắc lại mỗi từ đã được truyền đi, vì thế bản mã hóa trở thành:

Lệnh	**Stop**	**Play**	**FF**	**REW**
Từ mã	0000	0101	1010	1111

Bây giờ nếu ta truyền 1010 và một lỗi đơn giản xảy ra trong bit đầu tiên thì VCR nhận được 0010. Đây không phải là từ mã mà được gọi đơn giản là **từ** (word). Vì hai chữ số đầu tiên là 00 khác với cặp thứ hai là 10, VCR phát hiện một lỗi đã xảy ra trong quá trình truyền tin. Tuy nhiên VCR không thể

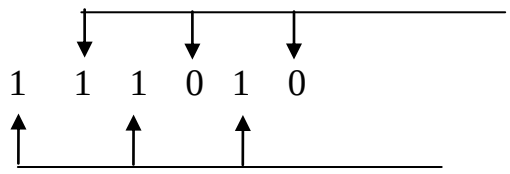
sửa lỗi này vì không có cách nào để biết 0010 là 1010 (lỗi trong bit đầu tiên) hoặc là 0000 (lỗi trong bit thứ 3). Mặc dầu vậy, đã có một sự cải tiến vì VCR không thực hiện được lệnh nhưng đã xác định được thông tin sai.

Ta cũng có thể thấy một dạng mã lặp trong công việc của người thu ngân trong siêu thị. Nếu vì một vài lý do nào đó máy quét đọc sai mã vạch, một thông tin "lỗi" sẽ được hiển thị trên máy kiểm tra và người thu ngân sẽ thực hiện lại thao tác để nhập mã, thông thường bằng cách dùng tay.

Nếu nhắc lại hai lần thì các từ mã sẽ là:

Lệnh	**Stop**	**Play**	**FF**	**REW**
Từ mã	000000	010101	101010	111111

Bây giờ nếu ta muốn truyền 101010 và lại một lỗi đơn xảy ra, thí dụ trong bit thứ hai là 1 và thông tin nhận được là 111010. Ta không chỉ phát hiện ra lỗi như trước đây mà còn có thể sửa lỗi. Điều này được làm bởi cách cách "đếm vượt" (majority count), đầu tiên cho mỗi cặp bit, như sau:



Bit này phải là 0

Những bit này là 1

Bit thứ hai phải là 0 bởi vì số 0 đã xuất hiện 2 trong 3 lần.

Vậy từ nhận được sẽ được giải mã là 101010.

Quá trình nhận được từ mã (codeword) theo **từ** (word) đã được chuyển đi được gọi là **giải mã** (decoding).

Trong ví dụ trên VCR có thể quyết định ngay rằng lệnh là 10 và là *tua đi*.

Mã lặp sửa những lỗi truyền đơn giản giống như trong ví dụ trên. Tuy nhiên nó cũng cho những khó khăn riêng là thông tin gốc được truyền đi ba lần. Điều này không những đắt mà còn có thể khó thực hiện vì, chẳng hạn dung lượng thông tin thường bị giới hạn bởi đường truyền.

Ta cũng thấy rằng, những thông tin bổ xung (thông tin lặp) bản thân nó cũng có thể bị lỗi, do đó việc giải mã có thể không được đảm bảo. Ví dụ tin nhắn nhận được ở trên 111010 có thể là 111111 với 2 lỗi ở bit thứ tư và thứ sáu.

Trong suốt chương này ta giả thiết rằng, nhờ sự tin cậy của thiết bị điện tử, xác suất xảy ra một lỗi đơn là rất nhỏ, đến mức xác suất xảy ra một lỗi đơn cũng chẳng khác hai hoặc nhiều hơn các lỗi như vậy. Mục tiêu của ta là làm tăng xác suất tin nhắn nhận được đúng càng cao càng tốt.

Ý tưởng này mở rộng thành khái niệm **giải mã lân cận gần nhất** (nearest-neighbour decoding). Người nhận có được một danh sách đầy đủ các từ mã. Nếu từ nhận được không phải là từ mã thì một hoặc nhiều các lỗi được phát hiện ngay. Để hiệu chỉnh lỗi, ta chọn từ mã giống nhất với từ đã được truyền đi bằng cách so sánh những từ nhận được với danh sách đã có và lựa chọn từ mã mà sai khác với từ đã nhận được với số lỗi nhỏ nhất. Ví dụ, với mã lặp 6 bit trên nếu nhận được 101111 thì so sánh với 4 mã từ cho ta:

Mã từ	000000	010101	101010	111111
Lỗi	101111	101111	101111	101111
Số lỗi	5	4	2	1

Tin nhắn vì vậy được giải mã là 1111111, vì đó là lân cận gần nhất với từ nhận được, chỉ sai khác 1 bit.

2.2. Mã chẵn lẻ

Ví dụ 2.2 (Mã cơ số 2-mã nhị phân)

Một mã đơn giản nhưng hữu ích cho việc sửa sai là mã nhận bởi cách nối thêm 1 bit đơn bổ xung vào mỗi tin nhắn chứa những thông tin được truyền đi. Bit này được lựa chọn sao cho từ mã kết quả chứa số chẵn chữ số 1. Điều này dẫn đến **mã chẵn lẻ** (even parity code). Tương tự, nếu bit bổ xung, được gọi là **bit kiểm tra chẵn lẻ** (parity-check bit), hoặc đơn giản là bit kiểm

tra (check bit), được lựa chọn để từ mã kết quả chứa số lẻ chữ số 1 thì mã kết quả được gọi là **mã lẻ** (odd parity code).

Chẳng hạn, xét mã đầu tiên trong ví dụ 2.1 và đặt hoặc số 0 hoặc số 1 vào cuối mỗi từ mã để mỗi từ mã mới là một số chẵn.

Lệnh	**Stop**	**Play**	**FF**	**REW**
Từ mã	000	011	101	111

Nếu, thí dụ, 101 được gửi đi và một lỗi đơn xảy ra thì từ nhận được sẽ là một trong các số 001, 111, 100. Mỗi từ này chứa số lẻ chữ số 1, vì thế lỗi đã được phát hiện. Tuy nhiên nó không thể xác định được bit nào cần được sửa.

Trong trường hợp tổng quát, giả sử tin nhắn gốc gồm n – 1 bit $x_1, x_2, ..., x_{n-1}$ được gọi là **các bit thông tin** (information bits). Một bit bổ xung x_n, được gọi là **bit kiểm tra**, được lựa chọn để từ mã là chẵn. Như vậy, các từ mã chứa một số chẵn các chữ số 1. Vì thế

$$x_1 + x_2 + ... + x_n \equiv 0 (\text{mod } 2). \tag{2.1}$$

Các từ mã được truyền đi là $x_1, x_2, ..., x_n$, trong đó x_i là 0 hoặc 1, và **độ dài** (length) của các từ mã, hoặc của bản thân mã, được định nghĩa là n. Giả sử một lỗi đơn xảy ra trong quá trình truyền tin ở bit thứ i. Điều này nghĩa là nếu $x_i = 0$ thì bit thứ i của từ nhận được là $y_i = 1$ và tương tự $y_i = 0$ nếu $x_i = 1$. Điều này có thể biểu diễn như sau:

$$y_i \equiv (x_i + 1) (\text{mod } 2) \tag{2.2}$$

và từ nhận được là $r = x_1 x_2 ... x_{i-1} y_i x_{i+1} ... x_n$. Tính chẵn lẻ của r nhận được bằng cách cộng các chữ số của nó theo đồng dư 2, được cho bởi

$$\begin{aligned} & x_1 + x_2 + ... + x_{i-1} + y_i + x_{i+1} + ... + x_n \\ & \equiv x_1 + x_2 + ... + x_{i-1} + (x_i + 1) + x_{i+1} + ... + x_n (\text{mod } 2) \\ & \equiv 1 (\text{mod } 2). \end{aligned}$$

Điều này chỉ ra rằng r có tính lẻ. Ta có thể dễ dàng thấy rằng trong trường hợp tổng quát, nếu một số lẻ của lỗi truyền tin xảy ra thì tính chẵn lẻ của $r=1$. Nói cách khác, nếu từ nhận được có tính lẻ thì đã xảy ra một số lẻ lỗi. Tương tự, nếu từ nhận được có tính chẵn thì ở đó phải có một số chẵn các lỗi (hoặc không có lỗi).

Giả sử rằng xác suất của một lỗi đơn bất kì là rất nhỏ. Khi ấy nếu từ nhận được có tính lẻ thì hầu như chỉ có đúng một lỗi đã xảy ra; và nếu từ nhận được có tính chẵn thì hầu như không có lỗi. Hơn nữa, với giả thiết ở trên, mã kiểm tra tính chẵn phát hiện mọi lỗi đơn. Nhưng nó không thể sửa chữa lỗi đơn bởi vì không có cách xác định bit nào trong từ nhận được là sai.

Xét một ví dụ khác, giả sử tính chẵn của mã cần kiểm tra có độ dài 5 đã được sử dụng và tin được gửi đi là 0111. Để thỏa mãn (2.1), bit kiểm tra phải là x_5= 1, vì thế từ mã được truyền đi là 01111 có tính chẵn. Giả sử tối thiểu có một lỗi trong quá trình truyền tin, khi ấy nếu nhận được 01111, thì từ này có thể sửa được bởi vì nó có tính chẵn. Sau khi bỏ bit kiểm tra, giải mã tin nhắn là 0111. Tuy nhiên, nếu từ nhận được, thí dụ, là 10101, thì vì nó có tính lẻ, nên một lỗi đã xảy ra, nhưng không xác định được bit nào trong từ nhận được là sai. Từ nhận được được giải mã bằng cách thông báo ‘lỗi’.

Một ví dụ khác. Xét mã 7 bit ASCII. Nó thường được mở rộng cho mã 8 bit bằng cách nối thêm 1 bit kiểm tra chẵn. Từ mã 7 bit 1000001 cho chữ *A* chở thành 1000010 và một vài từ mã 8 bit được cho trong bảng dưới đây:

Kí tự	A	B	C	a	b	c
Từ mã	10000010	10000100	10000111	11000011	110000101	01100011

Ví dụ 2.3 (Mã cơ số 3-mã tam phân)

Trong mã tam phân các từ mã là $x_1x_2...x_{n-1}x_n$, trong đó x_i là 0, 1 hoặc 2. Số kiểm tra x_n được chọn sao cho:

$$x_1+x_2+...+x_n \equiv 0(\text{mod } 3) \qquad (2.3)$$

Lỗi e_i trong chữ số x_i bất kì có thể là 1 hoặc 2. Nói riêng, nếu chỉ có một lỗi đơn ở chữ số thứ *i*, thì từ nhận được là $z = x_1x_2...x_{i-1}y_ix_{i+1}...x_n$, trong đó $y_i \equiv (x_i + e_i) (\text{mod } 3)$.

Trong trường hợp này, tổng các số của z là

$$x_1 + x_2 + ... + x_{i-1} + y_i + x_{i+1} + ... + x_n$$

$$\equiv x_1 + x_2 + ... + x_{i-1} + (x_i + e_i) + x_{i+1} + ... + x_n (\text{mod } 3) \equiv e_i (\text{mod } 3)$$

Vì thế, giả sử có một lỗi đơn xảy ra trong truyền tin thì mã phát hiện tất cả các lỗi đó bởi vì tổng các chữ số của từ nhận được không chia hết cho 3.

Thí dụ, xét thông tin gửi đi là 10210. Chữ số kiểm tra x_6 được chọn sao cho (2.3) được thỏa mãn, nghĩa là $1+0+2+1+0+x_6 \equiv 0 (\text{mod } 3)$. Suy ra x_6= 2. Do đó, từ mã được truyền đi là 102102. Nếu từ nhận được là 112102 thì tổng các số của nó là $1+1+2+1+0+2 \equiv 1 (\text{mod } 3)$, chỉ ra rằng một lỗi đã xảy ra.

Đúng hơn, từ "chẵn lẻ" (parity) chỉ áp dụng cho "tính chẵn" (even-ness), hoặc "tính lẻ" (odd-ness). Mã chẵn lẻ được mở rộng tự nhiên cho mã với cơ số p bất kì với tổng các chữ số của mỗi từ đồng dư với 0 theo môđun p.

2.3. Mã vạch

Ví dụ 2.4 (Số hiệu hàng hóa Châu Âu-European Article Number)

Mã vạch trong Hình 1.1 là ví dụ của mã nhận dạng hàng hóa được sử dụng rộng rãi và được gọi là **EAN** (European Article Number), được công nhận là chuẩn vào năm 1976. Nó đã được phát triển từ mã sản phẩm chung **UPC** (Universal Product Code) được sử dụng ở Mỹ từ năm 1973.

Xét Hình 1.1. Mỗi số bao gồm 13 chữ số thập phân. Hai hay ba số đầu tiên là phân vùng quốc gia, chẳng hạn 30 là Pháp, 49 là Nhật, 50 là Anh, 80 là Italia, 888 là Singgapor, 885 là Thái Lan … và 893 là Việt Nam. Bốn hoặc

năm số tiếp theo là các số của nhà sản xuất, và ta có thể thấy trong Bảng 2.1, thí dụ 4980 đồng nhất với nhà xuất bản giáo dục và 6009 là công ty thuốc lá Sài Gòn. Năm số kế tiếp đưa ra con số duy nhất xác định các sản phẩm, chẳng hạn 42679 và 05015 tương ứng cho sách số học và thuốc lá Vinataba. Chữ số cuối cùng là số kiểm tra mà máy tính lưu trữ có thể kiểm tra xem số trên sản phẩm có đồng nhất với số của nhà máy không.

Bảng 2.1. Một số EAN

Sản phẩm	Mã hàng
Sách Số học của NXB GD	8934980426791
Vinataba của CT thuốc lá SG	8936009050154
Vở ghi Hải Tiến	8936014823033
Tesco plain flour	5000119101594
Abbot Ale	5010549000213
Maille Provencale mustard	3036817800295
Sacla pesto sauce	8001060375109
Book in figure	9780138340940

Chú ý rằng, giá của sản phẩm không phải một phần của mã vạch. Thông tin này lưu trong bộ nhớ máy tính của cửa hàng và thông báo cho máy thu ngân điện tử giá của mặt hàng mà mã vạch của nó đã quét được.

Mã EAN cho văn hóa phẩm (ISBN) được bắt đầu với chữ số 978 hoặc 979. Chín chữ số tiếp theo là chín chữ số đầu tiên của ISBN (Hình 1.2).

Nếu một EAN được kí hiệu bởi $x_1x_2...x_{11}x_{12}x_{13}$, trong đó x_i là các chữ số thập phân, thì chữ số kiểm tra x_{13} là được tính sao cho tổng kiểm tra:

$$S = x_1 + x_3 + x_5 + x_7 + x_9 + x_{11} + 3\left(x_2 + x_4 + x_6 + x_8 + x_{10} + x_{12}\right) + x_{13} \qquad (2.4)$$

là bội của 10, nghĩa là S thỏa mãn phương trình kiểm tra:

$$S \equiv 0 \left(\mathrm{mod}\, 10\right). \qquad (2.5)$$

Ví dụ, thành phần đầu trong Bảng 2.1 khi thay thế vào (2.4) cho ta:

$$S = 8+ 3 + 9 + 0 + 2 + 7 + 3(9 + 4 + 8 + 4 + 6 + 9) + 1 = 150 \equiv 0(\mathrm{mod}\,10).$$

Có thể dễ dàng kiểm tra các dãy khác trong Bảng 2.1 luôn thỏa mãn (2.5).

Nếu một lỗi trong đại lượng *e* xảy ra tại một chữ số x_i nào đó thì *S* trong (2.4) thay đổi thành *e* hoặc 3*e* tùy theo chỉ số *i* là chẵn hay lẻ. Trong trường hợp khác, vì $(0 \le e \le 9)$, sự thay đổi trong *S* sẽ không còn đồng dư với 0 theo môđun 10, vì thế (2.5) sẽ không thỏa mãn. Do đó mã EAN phát hiện mọi lỗi đơn, tuy nhiên không sửa được các lỗi đơn vì không biết chữ số nào của EAN là sai.

Một loại lỗi phổ biến xảy ra khi nhập các chữ số từ bàn phím, hoặc khi đọc chúng, là hai chữ số kề nhau bị vô tình đổi chỗ. Nếu, chẳng hạn, khi đó trong chữ số thứ năm x_5 và thứ sáu x_6 ở một EAN được đổi chỗ, thì trong tổng kiểm tra (2.4) x_5 được thay thế bởi x_6 và $3x_6$ bởi $3x_5$. Sự thay đổi trong tổng kiểm tra khi đó là $-x_5 + x_6 - 3x_6 + 3x_5 = 2(x_5 - x_6)$. Do đó nếu $x_5 - x_6 = \pm 5$ thì sự thay đổi trong tổng kiểm tra sẽ là ± 10, vì thế tổng kiểm tra sẽ đồng dư với 0 theo môđun 10 và lỗi sẽ không bị phát hiện. Rõ ràng điều này có thể áp dụng tương tự đối với sự đổi chỗ của hai chữ số liền kề bất kỳ sai khác bởi 5 đơn vị (không nhất thiết phải là vị trí x_5 và x_6). Tuy nhiên, tất cả các lỗi này khi đổi chỗ hai chữ số liền kề sẽ được phát hiện. Chẳng hạn, nếu mã cho sản phẩm thuốc lá Vinataba trong Bảng 2.1 đọc không chính xác là 8934980426791 (những chữ số $x_{10} = 7$ và $x_{11} = 6$ đã được đổi chỗ) thì tổng kiểm tra (2.4) trở thành

$$S = 8+ 3 + 9 + 0 + 2 + 6 + 3(9 + 4 + 8 + 4 + 7 + 9) + 1 = 151$$

và vì vậy $S \equiv 1(\mathrm{mod}\,10)$, lỗi đã xảy ra. Tuy nhiên nếu, chẳng hạn mã cho Vinataba của công ty thuốc lá Sài Gòn đọc không đúng là 8936009500154 thì tổng kiểm tra (2.4) trở thành:

S = 8 + 3 + 0 + 9 + 0 + 1 + 3(9 + 6 + 0 + 5 + 0 + 5) + 4 = 100 $\equiv 0 \pmod{10}$.

Vì thế sự đổi chỗ của chữ số thứ thứ bảy và thứ tám ($x_7 = 0$, $x_8 = 5$, $x_6 - x_7 = -5$) không được phát hiện.

Dạng rút ngắn 12 chữ số và 8 chữ số hình thức của EAN cũng được sử dụng, và một số chuỗi cửa hàng bán lẻ cũng có hệ thống mã vạch riêng cho mình.

Ví dụ 2.5 (Thư chuyển tiền của bưu điện Mỹ)

Thư chuyển tiền phát hành bởi hệ thống dịch vụ bưu điện nước Mỹ (USPS) đồng nhất với số $x_1x_2...x_{10}x_{11}$ với x_i là các chữ số thập phân. Số kiểm tra x_{11} được định nghĩa như là phần dư theo môđun 9 của số gồm 10 chữ số, đó là:

$$x_1 x_2 \ldots x_9 x_{10} \equiv x_{11} \pmod{9}. \qquad (2.6)$$

Do đó $0 \le x_{11} \le 8$. Chẳng hạn, số 10 chữ số 3844809642 chia cho 9, ta được:

3844809642 = 9 x 427201071 + 3.

Vì vậy, x_{11}= 3 và mã từ là 38448096423. Một cách tính toán đơn giản số kiểm tra x_{11} là thay (2.6) bởi đẳng thức đồng dư tương đương:

$$\sum_{i=1}^{10} x_i \equiv x_{11} \pmod{9}. \qquad (2.7)$$

Sử dụng (2.7) với số có 10 chữ số ở trên cho ta

$$3 + 8 + 4 + 4 + 8 + 0 + 9 + 6 + 4 + 2 = 48 \equiv 3 \pmod{9}.$$

Chứng tỏ rằng x_{11}= 3, như trên.

Nếu một lỗi xảy ra tại một chữ số nào đó trong mười chữ số đầu thì dễ dàng thấy rằng phương trình đồng dư (2.7) sẽ không còn đúng, trừ khi 0 được thay thế bởi 9 và 9 bởi 0. Vậy, mã phát hiện tất cả các lỗi đơn trong các chữ số x_1 đến x_{10}, trừ hai trường hợp trên. Chẳng hạn, nếu x_6 = 0 trong số có 10 chữ số ở ví dụ trên bị thay bởi 9 thì tổng ở vế trái trong (2.7) của số mới này

sẽ là 48 + 9 = 57 $\equiv 3(\text{mod } 9)$. Điều này chỉ ra rằng số kiểm tra không thay đổi, vì thế lỗi trong chữ số thứ sáu không bị phát hiện.

Nếu lỗi chỉ xảy ra ở chữ số kiểm tra x_{11} thì nó sẽ bị phát hiện vì (2.7) sẽ vi phạm. Trên thực tế khoảng 2% số các lỗi đơn sẽ không bị phát hiện.

Tuy nhiên, khả năng của mã phát hiện ra các lỗi ở trong việc đổi chỗ giữa hai chữ số thập phân x_i và x_{i+1} là rất kém. Thật vậy, nếu $i \le 9$ và hai chữ số liền kề đổi chỗ thì tổng bên vế trái trong (2.7) không thay đổi và do đó chữ số kiểm tra không thay đổi. Lỗi không bị phát hiện.

Khả năng còn lại có thể là khi x_{10} và x_{11} đổi chỗ. Trong trường hợp này tổng mới của các chữ số trong vế trái của (2.7) là:

$$x_1 + x_2 + ... + x_9 + x_{11} = x_1 + x_2 + ... + x_9 + x_{10} + (x_{11} - x_{10})$$

$$\equiv x_{11}(\text{mod } 9) + (x_{11} - x_{10})(\text{mod } 9)$$

$$\equiv x_{10}(\text{mod } 9) + (2x_{11} - 2x_{10})(\text{mod } 9).$$

Biểu thức cuối cùng này không thể đồng dư với $x_{10}(\text{mod } 9)$ trừ khi $x_{10} = x_{11}$, bởi vì $x_{11} \le 8$ nên $(2x_{11} - 2x_{10}) \not\equiv 0(\text{mod } 9)$. Điều này chỉ ra rằng từ $x_1x_2...x_{10}x_{11}$ không phải là một từ mã, trừ khi $x_{10} = x_{11}$, trong trường hợp đó sự đổi chỗ của x_{10} và x_{11} là không phân biệt. Vì vậy mã chỉ phát hiện các lỗi truyền khi chúng chứa chữ số kiểm tra.

§ 3. Khoảng cách Hamming

Bốn lệnh trong ví dụ 2.1 được biểu diễn bởi 00, 01, 10, 11. Nếu có một lỗi đơn trong truyền tin thì lệnh sai sẽ được thực hiện. Đó là bởi vì bốn lệnh 00, 01, 10, 11 chứa toàn bộ các khả năng có thể của tổ hợp hai số 0 và 1, do đó một lỗi trong một bit của một từ mã sẽ chuyển nó thành một từ mã khác – các từ mã đó "quá gần nhau". Ý tưởng này đã được khái quát hóa bởi nhà toán học và khoa học máy tính người Mỹ R.W. Hamming năm 1950.

Kí hiệu $\mathbb{Z}_p$ là tập hợp các số nguyên 0, 1, 2, …, $p - 1$ (đại diện của các lớp đồng dư môđun p). Một xâu các chữ số $x = x_1x_2...x_{n-1}x_n$, trong đó mỗi x_i thuộc $\mathbb{Z}_p$ được gọi là **một từ** có **độ dài** n. Vì mỗi x_i có thể nhận p giá trị cho nên tổng cộng có p^n từ. Một ***p*-mã** C chiều dài n là một tập hợp các từ như vậy, và nếu $x \in C$ thì x được gọi là một **từ mã**.

Trường hợp riêng, nếu $p = 2$ thì x_i bằng 0 hoặc 1 và C là một **mã nhị phân**. Nếu $p = 3$, thì x_i bằng 0, 1 hoặc 2 và C là **mã tam phân**.

Trong chương này ta chỉ xét các mã mà tất cả các từ mã có cùng độ dài. Giả sử x và y là hai từ có độ dài n. Khi ấy **khoảng cách Hamming** d(x, y) giữa chúng là số các vị trí mà x và y là khác nhau.

Thí dụ, với mã tam phân có độ dài 5 thì d(01221, 10211) = 3 vì hai từ này khác nhau tại các vị trí thứ 1, 2 và 4.

Có thể thấy d(x, y) chính là số nhỏ nhất các chữ số của x phải thay đổi để nhận được y.

Thuật ngữ "khoảng cách" được sử dụng, bởi vì d(x, y) thỏa mãn ba tiên đề của khoảng cách. Hai tiên đề đầu tiên suy ra ngay từ định nghĩa:

(1) d(x, y) = 0 nếu và chỉ nếu $x = y$, nghĩa là $x_i = y_i$ với i = 1, 2, …., n. Nếu ngược lại thì d(x, y) > 0.

(2) d(x, y) = d(y, x).

(3) Tiên đề bất đẳng thức tam giác:

Với bất kì từ thứ ba nào $z = z_1z_2...z_n$, ta có:

$$d(x, y) \le d(x, z) + d(z, y). \qquad (3.1)$$

Để chứng minh (3.1), giả sử x thay đổi thành y bằng cách đi qua z. Số chữ số thay đổi từ x đến z là d(x, z) và từ z đến y là d(z, y), cho ta tổng chữ số thay đổi: t = d(x, z) + d(z, y). Tuy nhiên, d(x, y) là số nhỏ nhất có thể các chữ số cần thay đổi để x trở thành y, và vì thế không thể vượt quá t, bất đẳng thức (3.1) được thỏa mãn.

Khái niệm giải mã lân cận gần nhất giới thiệu ở ví dụ 2.1 bây giờ có thể được giải thích lại như sau: coi từ mã đã truyền đi là từ mã gần nhất với từ nhận được, theo nghĩa khoảng cách Hamming giữa hai từ là nhỏ nhất.

Một tham số quan trọng ảnh hưởng đến việc phát hiện lỗi và tính chất sửa sai của một mã C là sự "gần gũi toàn bộ" (overall closeness) của các từ, được đo bằng **khoảng cách tối thiểu** $\delta(C)$. Nó được định nghĩa như là giá trị nhỏ nhất của các d(x, y) với mọi x, y trong C với $x \neq y$.

Ví dụ 3.1 (Khoảng cách bé nhất cho ví dụ 2.1)

Mã đầu tiên được sử dụng trong ví dụ 2.1 là $C_1 = \{00, 01, 10, 11\}$, và khoảng cách giữa tất cả các cặp có thể có của từ mã là:

d(00, 01) = 1, d(00, 10) = 1, d(00, 11) = 2,

d(01, 10) = 2, d(01, 11) = 1, d(10, 11) = 1.

Do đó khoảng cách tối thiểu là $\delta(C_1) = 1$, và có thể giải thích lý do tại sao mã C_1 không thể phát hiện những lỗi đơn. Thật vậy, đối với bất kỳ mã C với $\delta(C) = 1$ sẽ có ít nhất hai từ mã a và b mà d(a, b) = 1. Nếu a được truyền đi với một lỗi a_i bị thay đổi thành b_i, khi ấy ta nhận được từ b. Nhưng do d(a,b)=1 nên b cũng là từ mã, tin nhắn được coi là đúng, vì vậy không có cách phát hiện lỗi đã xảy ra.

Bây giờ xét mã thứ hai trong ví dụ 2.1, trong đó mỗi tin nhắn được truyền hai lần, tạo thành $C_2 = \{a, b, c, e\}$ với $a = 0000$, $b = 0101$, $c = 1010$, $e = 1111$.

Khoảng cách Hamming giữa các mã từ bây giờ là d(a, b) = 2, d(a, c) = 2, d(a, e) = 4, d(b, c) = 4, d(b, e) =2, d(c, e) = 2. Do đó khoảng cách tối thiểu là $\delta(C_2) = 2$, và do đó mã C_2 có thể phát hiện tất cả các lỗi đơn.

Mã lặp thứ ba trong ví dụ 2.1 là $C_3 = \{a_1, a_2, a_3, a_4\}$ với $a_1 = 000000$, a_2=010101, a_3=101010, a_4=111111. Khoảng cách giữa các từ mã là

d(a_1, a_2)=3, d(a_1, a_3)=3, d(a_1, a_4)=6, d(a_2, a_3)=6, d(a_2, a_4)=3, d(a_3, a_4)=3 và khoảng cách tối thiểu là $\delta(C_3)$ = 3. Do đó mã này có thể sửa được tất cả các lỗi đơn.

Ví dụ này minh họa kết quả quan trọng dưới đây, chỉ ra tại sao khoảng cách tối thiểu xác định được lỗi và các tính chất sửa sai của một mã.

Định lý 3.1 (Phát hiện và sửa lỗi)

Giả sử *C* là một mã có khoảng cách tối thiểu là δ, và giả thiết những từ nhận được được giải mã bằng nguyên lý lân cận gần nhất. Khi ấy:

(1) *C* sẽ phát hiện *e* lỗi với điều kiện

$$\delta \geq e+1. \tag{3.2}$$

(2) *C* sẽ sửa các lỗi *e* với điều kiện

$$\delta \geq 2e+1. \tag{3.3}$$

Chứng minh

(1) Theo định nghĩa của khoảng cách tối thiểu, mọi cặp từ mã khác nhau ở ít nhất δ vị trí. Theo (3.2), các cặp từ mã khác nhau ở ít nhất *e* + 1 vị trí. Giả sử một mã *x* đã được truyền đi và nhiều nhất *e* lỗi xảy ra trong quá trình truyền. Khi ấy, từ nhận được sẽ khác với *x* trong nhiều nhất là *e* vị trí, và vì vậy không phải là một từ mã. Do đó nhiều nhất *e* lỗi truyền được phát hiện.

(2) Giả sử một lần nữa rằng một từ mã *x* đã được truyền và nhiều nhất *e* lỗi xảy ra, do đó từ nhận được *z* khác với *x* ở tối thiểu *e* vị trí, nghĩa là,

$$d(x, z) \leq e. \tag{3.4}$$

Nếu *y* là từ mã bất kỳ, khác với từ mã *x*, khi đó theo định nghĩa của khoảng cách tối thiểu:

$$d(x, y) \geq \delta \geq 2e+1. \tag{3.5}$$

Thay thế (3.4) vào bất đẳng thức tam giác (3.1) cho ta:

$$d(x, y) \leq e + d(z, y).$$

Kết hợp điều này với (3.5) ta được

$$d(z, y) \geq d(x, y) - e \geq 2e + 1 - e = e + 1.$$

Điều này cho thấy khoảng cách giữa từ nhận được z và bất kỳ từ mã y ($\neq x$) tối thiểu là e+1. Do đó theo (3.4), từ mã gần nhất đối với z là x, vì thế theo nguyên lý lân cận gần nhất, từ nhận được được giải mã là x, tối đa e các lỗi truyền đi được sửa.

Các bất đẳng thức (3.2) và (3.3) được viết lại là $e \leq \delta - 1$, và $e \leq \frac{1}{2}(\delta - 1)$.

Định lý có thể phát biểu lại như sau:

Nếu một mã C có khoảng cách tối thiểu là δ, thì C có thể được sử dụng hoặc là để phát hiện tối đa δ-1 lỗi; hoặc để sửa nhiều nhất $\frac{1}{2}(\delta - 1)$ lỗi nếu δ là lẻ, hoặc $\frac{1}{2}(\delta - 2)$ nếu δ là chẵn.

Khi những lỗi được phát hiện, nhưng không thể sửa được thì người nhận yêu cầu gửi lại tin nhắn.

Ví dụ 3.2 (Những ứng dụng của định lý)

(a) Áp dụng định lý vào ba mã trong ví dụ 3.1 cho ta bảng dưới đây

Mã	δ	**Số lỗi được phát hiện**	**Số lỗi được sửa**
C_1	1	0	0
C_2	2	1	0
C_3	3	2	1

Các kết quả này phù hợp với khẳng định khi xét các mã lặp lại trong Ví dụ 2.1. Chú ý rằng bây giờ vì bổ xung thêm mã C_3 (khi tin nhắn được nhắc lại hai lần) nên nó có thể sửa một lỗi hoặc phát hiện hai lỗi, bởi vì khoảng cách giữa các từ mã ít nhất là 3.

(b) Xét một mã nhị phân có độ dài 5

$$C = \{11001,\ 01110,\ 10100,\ 00011\}$$

Dễ dàng thấy rằng khoảng cách tối thiểu là δ = 3. Mã hoặc có thể phát hiện tối đa hai lỗi, hoặc sửa được lỗi đơn. Để minh họa hai trường hợp này, trước hết giả sử rằng một từ $r = 10010$ là nhận được. Khoảng cách của nó với bốn từ mã là d(*r*, 11001) = 3, d(*r*, 01110) = 3, d(*r*, 10010) = 2, d(*r*, 00011) = 2. Vì các từ mã thứ ba và thứ tư là bằng nhau, theo nguyên lý lân cận gần nhất thì không thể xác định được từ mã nào được truyền đi, có nghĩa là, những lỗi không thể được sửa. Tuy nhiên, hai lỗi đã được phát hiện, do đó chẳng hạn *r* có thể là từ 10100 với những lỗi trong các bit thứ hai và thứ ba, hay từ 00011 với lỗi trong bit đầu tiên và cuối cùng.

Thí dụ khác, giả sử nhận được 11100. Những khoảng cách tới những từ mã tương ứng là 2, 2, 1, 5 chỉ ra rằng từ mã thứ ba là gần nhất tới từ nhận được. Theo nguyên lý lân cận gần nhất ta kết luận rằng từ mã thứ ba 10100 đã được truyền đi, qua đó sửa chữa lỗi đơn (trong bit thứ hai).

§4. Mã tuyến tính

4.1. Mã nhị phân tuyến tính

Định nghĩa *tổng* hai từ mã nhị phân $x = x_1x_2...x_n$ và $y = y_1y_2...y_n$ là

$$z = x + y \text{ mà } z_i \equiv x_i + y_i \pmod 2,\ i = 1, 2, \ldots, n.$$

Có nghĩa là, các bit được cộng theo quy tắc cộng môđun 2:

$$0 + 0 = 0, \qquad 1 + 1 = 0, \qquad 1 + 0 = 1 = 0 + 1. \tag{4.1}$$

Từ (4.1) ta có $z_i = 0$ nếu $x_i = y_i$, và $z_i = 1$ nếu $x_i \neq y_i$.

Mã nhị phân tuyến tính là một mã *C* có tính chất tổng của hai từ mã bất kỳ cũng là một từ mã.

Nghĩa là, nếu *x* và *y* thuộc *C* thì $z = x + y$ cũng thuộc *C*. Đặc biệt, nếu *x* = *y* thì bit thứ *i* của $x + x$ là $x_i + x_i = 1 + 1 = 0$, hoặc $x_i + x_i = 0 + 0 = 0$.

Điều này chỉ ra rằng mọi mã nhị phân tuyến tính bất kỳ luôn luôn chứa **từ mã 0** (có tất cả các bit đều là chữ số không).

Ví dụ 4.1 (Mã tuyến tính và mã phi tuyến)

(a) Xét mã nhị phân $C = \{000,\ 100,\ 101,\ 001\}$. Các bit của tổng 100 + 101 là 1 + 1 = 0, 0 + 0 = 0, 0 +1 = 1, do đó, 100 + 101 = 001 cũng thuộc *C*.

Tương tự, những tổng khác của những cặp từ mã là:

000 + 100 = 100, 000 + 101 = 101, 000 + 001 = 001,

100 + 001 = 101, 101 + 001 = 100.

Tất cả là từ mã, vì vậy *C* là mã tuyến tính.

(b) Mã nhị phân chẵn lẻ trong ví dụ 2.2 là một mã tuyến tính. Thật vậy, bất kỳ hai từ mã *x* và *y* thỏa mãn các điều kiện (2.1)

$\sum x_i \equiv 0, \sum y_i \equiv 0 (\text{mod } 2)$. Do đó tổng các bit của $z = x + y$ là:

$$\sum z_i = \sum (x_i + y_i) = \sum x_i + \sum y_i \equiv 0 (\text{mod } 2).$$

Điều này cho thấy rằng *z* cũng thỏa mãn (2.1), và do đó cũng là một từ mã.

(c) Mã *C* = (0000, 1010, 0111) là không tuyến tính, bởi vì:

1010+0111=1101 không phải là một từ mã.

Theo Định lý 3.1, ta cần phải tính khoảng cách tối thiểu δ cho một mã để xác định lỗi và xử lý lỗi đó. Một lý do mã tuyến tính quan trọng bởi vì giá trị của δ có thể tìm thấy dễ dàng bằng cách tính khoảng cách giữa tất cả các cặp có thể của các từ mã.

Ta đưa vào định nghĩa:

Trọng số w(*x*) của một từ mã nhị phân *x* là số các số 1 trong *x*.

Xét tổng *z* của hai mã từ *x* và *y*. Dễ thấy rằng $z_i = 1$ khi $x_i \neq y_i$, do đó, số các chữ số 1 trong *z* bằng với số những vị trí mà *x* và *y* khác nhau.

Do đó từ định nghĩa của d(*x*,*y*) suy ra w(*z*) = w(*x* + *y*) thỏa mãn:

$$w(x + y) = d(x, y). \qquad (4.2)$$

Đặt *y* = 0, (4.2) cho thấy rằng với mọi từ mã *x*, trọng số của nó được cho bởi:

$$w(x) = d(x, 0). \qquad (4.3)$$

Như vậy mối quan hệ giữa khoảng cách tối thiểu và trọng số được thiếtt lập bởi định lý dưới đây.

Định lý 4.1

Khoảng cách tối thiểu δ cho một mã nhị phân tuyến tính C bằng trọng số khác không nhỏ nhất của các từ mã, nghĩa là,

$$\delta = \min_{x \neq 0} \mathrm{w}(x). \tag{4.3}$$

Chứng minh

Ký hiệu w* là giá trị nhỏ nhất của w(x) trong (4.3), và để cho u là một từ mã mà w(u) = w*. Từ (4.2) suy ra $\mathrm{w}^* = d(u,0)$. Bởi vì cả hai u và 0 là từ mã nên từ định nghĩa khoảng cách tối thiểu ta cũng suy ra $\delta \leq$ d(u, 0). Kết hợp các kết quả này ta được $\delta \leq$ w*.

Mặt khác, $\delta \geq$ w*. Vậy $\delta = \mathrm{w}^*$, hay (4.3) được chứng minh.

Ví dụ 4.2 (Ứng dụng của định lý)

(a) Trọng số của các từ mã khác không trong mã 4.1a) là w(100)=1, w(101)=2, w(001) = 1. Do đó theo (4.3) thì khoảng cách tối thiểu là δ = 1.

(b) Dễ dàng kiểm tra rằng mã C = (000000, 010101, 101010, 111111) là tuyến tính. Các trọng số là w(010101) = 3, w(101010) = 3, w(111111) = 6, như vậy theo (4.3) mã có khoảng cách tối thiểu δ = 3.

4.2. Biểu diễn ma trận của các mã nhị phân

Ví dụ 4.3 (Phương trình kiểm tra)

Xét một mã nhị phân C_1 độ dài 3 mà các từ mã $x_1x_2x_3$ thỏa mãn điều kiện $x_1 + x_2 = 0$. Có tám từ nhị phân chiều dài 3 là 000, 001, 010, 011, 100, 101, 110, 111. Chọn từ 8 từ đó các từ thỏa mãn điều kiện trên thì ta có $C_1 = \{000, 001, 110, 111\}$. Vì $\delta(C_1)$ = 1 nên mã thậm chí không phát hiện được những lỗi đơn. Nếu thêm một điều kiện là $x_1 + x_3 = 0$ thì chỉ có từ mã đầu tiên và cuối cùng trong C_1 thỏa mãn điều kiện này, vì vậy mã thỏa mãn cả hai

điều kiện là C_2={000,111}. Rõ ràng $\delta(C_2)$=3, vì vậy mã này có thể sửa chữa mọi lỗi đơn.

Các phương trình trên được gọi là phương trình **kiểm tra tính chẵn lẻ** (hay đơn giản là phương trình kiểm tra).

Ví dụ này minh họa một ý tưởng tổng quát trong việc xây dựng mã tuyến tính. Mục tiêu là thêm các điều kiện dưới dạng của những phương trình kiểm tra tuyến tính nhằm làm tăng khoảng cách tối thiểu giữa các từ mã, và do đó có thể cải thiện khả năng xử lý lỗi. Nhưng điều này sẽ làm giảm bớt số lượng các từ mã, và bởi vậy làm giảm bớt khả năng truyền thông tin của mã. Chẳng hạn, C_1 chứa bốn tin nhắn nhưng chỉ có hai tin nhắn có thể được gửi bằng cách sử dụng C_2 (ví dụ, "có" và "không").

Nói chung, những điều kiện mà các bit của từ mã phải thỏa mãn có dạng phương trình tuyến tính, nên sử dụng ngôn ngữ của ma trận là tiện lợi. Nói một cách chặt chẽ, những "phương trình" là những phương trình biểu diễn đồng dư theo môđun 2, nhưng để tiện lợi ta biểu diễn chúng như những phương trình thông thường.

Một phương trình kiểm tra

$$a_1x_1 + a_2x_2 + ... + a_nx_n = 0$$

có thể được viết dưới dạng thu gọn $ax = 0$, với $a = [a_1, a_2, ..., a_n]$ là véctơ hàng với thành phần là $a_1, a_2, ..., a_n$; x là véctơ mã

$$x = [x_1, x_2, ..., x_n]^T \tag{4.4}$$

và

$$ax = a_1x_1 + a_2x_2 + ... + a_nx_n \tag{4.5}$$

là tích vô hướng của a và x, T là kí hiệu chuyển vị ma trận.

Chú ý rằng véctơ mã (4.4) được viết như một cột với những chữ số của từ mã $x = x_1x_2...x_n$ đọc từ trái sang phải và được đặt từ trên xuống dưới.

Nếu có nhiều phương trình kiểm tra, thì chúng có thể được viết dưới dạng:

$$\begin{bmatrix} a_1 & a_2 & \dots & a_n \\ b_1 & b_2 & \dots & b_n \\ . & . & \dots & . \\ . & . & \dots & . \end{bmatrix} x = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \text{ hay } Hx = 0.$$

Ở đây H là **ma trận kiểm tra**, các hàng của nó là các hệ số của các phương trình kiểm tra, và 0 là vectơ không.

Chú ý rằng với các mã nhị phân, tất cả các thành phần trong H là 0 hoặc 1, và H được gọi là một ma trận nhị phân.

Ví dụ 4.4 (Ma trận kiểm tra)

Sử dụng (4.5), hai phương trình kiểm tra trong ví dụ 4.3, cụ thể là

$$1x_1 + 1x_2 + 0x_3 = 0, \qquad 1x_1 + 0x_2 + 1x_3 = 0$$

có thể được viết như những tích vô hướng

$$\begin{bmatrix} 1 & 1 & 0 \end{bmatrix} x = 0, \quad \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} x = 0, \tag{4.6}$$

trong đó $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$.

Kết hợp hai biểu thức trong (4.6) cho ta

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} x = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0. \tag{4.7}$$

Biểu thức (4.7) có thể được viết gọn như sau

$$Hx \equiv 0 \pmod 2. \tag{4.8}$$

Ma trận kiểm tra là ma trận nhị phân

$$H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}. \tag{4.9}$$

Từ mã $x_1x_2x_3$ được xác định như là nghiệm của các phương trình kiểm tra (4.8), nó đã được tìm thấy trong ví dụ 4.3 là 000 và 111.

Giả sử x và y là hai từ mã trong một mã C thỏa mãn cùng một hệ phương trình kiểm tra dưới dạng ma trận:

$$Hx = 0, \quad Hy = 0.$$

Nếu $z = x + y$, thì

$$Hz = H(x + y) = Hx + Hy = 0 + 0 = 0.$$

Điều này chỉ ra rằng z cũng thỏa mãn hệ phương trình kiểm tra, và do đó cũng thuộc C. Vì vậy, theo định nghĩa, C là một mã tuyến tính. Nói cách khác, một mã nhị phân tuyến tính C là tập các từ mã thỏa mãn điều kiện

$$Hx \equiv 0 (\text{mod } 2), \tag{4.10}$$

trong đó x là véctơ mã.

Nếu có m phương trình kiểm tra thì H trong (4.10) có m hàng và n cột, và được gọi là có **kích thước** $m \times n$, hay là một ma trận $m \times n$.

Ví dụ 4.5 (Tạo một mã tuyến tính)

Những ví dụ 4.3 và 4.4, các từ mã đã nhận được bằng cách đơn giản là tìm trong tập tất cả các từ có chiều dài 3 các từ thỏa mãn các phương trình kiểm tra (4.8). Phương pháp này sẽ phức tạp nếu n quá lớn. Dưới đây mô tả một cách tìm các từ mã cho trường hợp n bất kì.

Giả sử ma trận kiểm tra là

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}. \tag{4.11}$$

Những phương trình kiểm tra tương ứng với (4.11) nhận được bằng cách viết các hàng của H như những hệ số của phương trình. Chẳng hạn, hàng đầu tiên của H cho $1x_1 + 0x_2 + 1x_3 + 0x_4 = 0,$ hoặc

$$x_1 + x_3 = 0. \tag{4.12}$$

Tương tự dòng thứ hai của H cho

$$x_2 + x_3 + x_4 = 0. \tag{4.13}$$

Các từ mã là các nghiệm của (4.12) và (4.13). Phép cộng nghịch đảo trong $\mathbb{Z}_2$ là – 0 = 0, -1 = 1, do đó trong $\mathbb{Z}_2$ cũng cho ta: $-x_i = x_i$. Phương trình (4.12) có thể được viết lại như sau:

$$x_1 = -x_3 = x_3 \tag{4.14}$$

và (4.13) có dạng

$$x_2 = -x_3 - x_4 = x_3 + x_4 . \tag{4.15}$$

Các bit x_1 và x_2 bây giờ đã được biểu thị dưới dạng x_3 và x_4 được coi là hai bit độc lập, và chúng nhận giá trị 0 hoặc 1. Tất cả có bốn khả năng sau

x_1	x_2	x_3	x_4
0	0	0	0
0	1	0	1
1	1	1	0
1	0	1	1

Đối với mỗi cặp giá trị của x_3 và x_4, giá trị tương ứng của x_1 và x_2 trong hai cột đầu tiên được chọn theo công thức (4.14) và (4.15). Ví dụ $x_3 = 1$, $x_4 = 0$ thì $x_1 = x_3 = 1$, $x_2 = x_3 + x_4 = 1$.

Từ bảng này có thể thấy rằng có bốn từ mã 0000, 0101, 1110, 1011. Sử dụng (4.3), khoảng cách tối thiểu là δ = min[w(0101), w(1110), w(1011)] = 2.

Vì x_3 và x_4 có thể chọn tùy ý nên chúng được gọi là các **bit thông tin**. Hai bit x_1 và x_2 được gọi là các **bit kiểm tra**, và các bit này được xác định duy nhất bởi các bit thông tin, như được chỉ ra trong bảng trên. Thông tin tin nhắn $x_3\, x_4$ được gọi là **mã hóa** (encoded) thành từ mã $x_1\, x_2\, x_3\, x_4$.

Ví dụ trên minh họa bài toán xây dựng một mã tuyến tính được thực hiện bằng cách chọn một ma trận kiểm tra phù hợp. Chú ý rằng các phương trình (4.12) và (4.13) dễ dàng giải được vì bit kiểm tra x_1 chỉ xuất hiện trong phương trình đầu tiên và bit kiểm tra x_2 chỉ xuất hiện trong phương trình thứ hai. Tương tự, ta xét mã có ba bit kiểm tra cho một mã chiều dài 5, với các phương trình kiểm tra dạng:

$$\begin{aligned} x_1 + \qquad & a_1x_4 + a_2x_5 = 0; \\ x_2 + \qquad & b_1x_4 + b_2x_5 = 0; \\ x_3 + & c_1x_4 + c_2x_5 = 0, \end{aligned} \tag{4.16}$$

trong đó x_4 và x_5 là những bit thông tin. Như ví dụ 4.1, hệ có thể được viết lại bằng cách sử dụng môđun 2 để biểu diễn các mã kiểm tra như sau

$$\left.\begin{aligned} x_1 &= a_1x_4 + a_2x_5 \\ x_2 &= b_1x_4 + b_2x_5 \\ x_3 &= c_1x_4 + c_2x_5 \end{aligned}\right\}. \tag{4.17}$$

Trong ký hiệu ma trận, (4.16) trở thành $Hx = 0$, trong đó:

$$H = \begin{bmatrix} 1 & 0 & 0 & a_1 & a_2 \\ 0 & 1 & 0 & b_1 & b_2 \\ 0 & 0 & 1 & c_1 & c_2 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix}^T. \tag{4.18}$$

Ba cột đầu tiên của H trong (4.18) tạo thành ma trận đơn vị 3×3, được ký hiệu là I_3. Các cột còn lại của (4.18) tạo thành một ma trận A số chiều 3 × 2.

Tương tự, ma trận kiểm tra (4.11) có thể được viết là $H = \begin{bmatrix} I_2 A \end{bmatrix}$ với:

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Tổng quát, để xây dựng một mã nhị phân tuyến tính chiều dài n với m bit kiểm tra, ma trận kiểm tra là: $H = \begin{bmatrix} I_m A \end{bmatrix}$, với A là ma trận có số chiều $m \times (n-m)$. Các từ mã $x = x_1x_2...x_n$ thỏa mãn phương trình kiểm tra $Hx = 0$. Các bit kiểm tra $x_1x_2...x_n$ được tính như sau:

$$\begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = A \begin{bmatrix} x_{m+1} \\ x_{m+2} \\ \dots \\ x_{m+n} \end{bmatrix}. \tag{4.19}$$

Chẳng hạn, (4.17) được viết thành:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = A \begin{bmatrix} x_4 \\ x_5 \end{bmatrix}, \text{ trong đó } A = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \\ c_1 & c_2 \end{bmatrix}.$$

Có tất cả $k = n - m$ bit thông tin $x_{m+1,}...,x_n$. Bởi vì các bit này có thể nhận giá trị 0 hay 1 độc lập, tổng cộng có tất cả 2^k từ mã. Số k được gọi là **số chiều** của mã, còn mã được gọi là một $[n,k]$ mã. Để mã hóa một tin nhắn chứa $x_{m+1,}...,x_n$, đơn giản ta thêm vào đầu nó những bit kiểm tra $x_1x_2...x_m$ được tính bằng cách sử dụng (4.19).

Ví dụ 4.6 (Xây dựng một mã tuyến tính)

Để xây dựng mã chiều dài 7 với ba bit kiểm tra, ta chọn ma trận kiểm tra

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} I_3 & A \end{bmatrix}. \qquad (4.20)$$

Các hàng của ma trận A trong (4.20) đơn giản là những hệ số của những phương trình trong vế phải của (4.19). Vì m = 3, các bit kiểm tra là x_1, x_2, x_3, do đó, ví dụ, dòng đầu tiên của A trong (4.20) cho $x_1 = 1x_4 + 1x_5 + 0x_6 + 1x_7$, có nghĩa là: $x_1 = x_4 + x_5 + x_7$.

Tương tự, dòng thứ hai và thứ ba của A trong (4.20) cho

$$x_2 = x_4 + x_5 + x_6, \qquad x_3 = x_5 + x_6 + x_7.$$

Như trong ví dụ 4.5, lựa chọn tất cả các giá trị có thể có của các bit thông tin x_4, x_5, x_6, x_7 cho tập tất cả các từ mã. Ví dụ, nếu x_4=1, x_5=0, x_6 = 1, x_7 = 0 thì:

$$x_1 = 1 + 0 + 0 = 1, \qquad x_2 = 1 + 0 + 1 = 0, \qquad x_3 = 0 + 1 + 0 = 1.$$

Vì vậy, từ mã tương ứng là 1011010.

Mã chiều dài 7 xây dựng như trên có kích thước là $k = 7 - 3 = 4$, và có tất cả $2^4 = 16$ từ mã. Bốn từ mã khác với 1011010 được liệt kê trong bảng sau.

Các bit kiểm tra			**Các bit thông tin**			
x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	1	0	1	0	0	0
1	1	1	0	1	0	0
0	1	1	0	0	1	0
0	0	1	1	1	0	0

Phát hiện lỗi đơn

Đối với một mã nhị phân, một lỗi đơn có nghĩa là một bit 0 được nhận là 1, hoặc bit 1 được nhận là 0. Theo Định lý 3.1, để mã phát hiện tất cả các lỗi đơn khoảng cách tối thiểu δ ít nhất phải là 2. Tuy nhiên theo Định lý 4.1, đối với một mã nhị phân tuyến tính, δ bằng số nhỏ nhất của tất cả các trọng số của từ mã khác không. Vì vậy trong trường hợp này đòi hỏi $\delta \geq 2$, suy ra rằng không có từ mã nào có trọng số 1. Nếu e là từ có trọng số 1 thì nó có mọi bit bằng 0 trừ một bit thứ i, tức là $e = [0...010...0]^T$. Từ e như vậy không là từ mã thì nó phải thỏa mãn $He \neq 0$.Tuy nhiên, vectơ tích He chính là cột thứ i của H, vì vậy suy ra không có cột nào của H có thể bằng không. Nói cách khác, ma trận kiểm tra H của mã tuyến tính nhị phân phát hiện lỗi đơn (single-error-detecting) không có cột nào bằng 0.

Ví dụ 4.7 (Hai ma trận kiểm tra)

(a) Ma trận kiểm tra H trong (4.11) không có cột số nào có tất cả các phần tử bằng không và do đó tạo ra một mã phát hiện tất cả các lỗi đơn. Điều này có được do $\delta = 2$ cho mã này.

(b) Ma trận kiểm tra $H = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$ có cột thứ hai bằng 0. Do đó 0100, với chỉ bit thứ hai khác không, là một từ mã có trọng số 1. Khoảng cách tối thiểu do đó là $\delta = 1$, và mã với ma trận kiểm tra H không nhận diện được những lỗi đơn.

Sửa chữa lỗi đơn

Trong §3 ta biết rằng, để sửa được những lỗi đơn thì $\delta \geq 3$, nghĩa là không có từ mã nào có trọng số 1 hoặc 2. Nếu e là một từ nhị phân có trọng số 2 thì e chỉ có hai bit khác không tại vị trí i và j và khi ấy $He \neq 0$. Tuy nhiên, điều này dẫn đến các cột h_i, h_j thứ i và thứ j của H thỏa mãn điều kiện $h_i + h_j \neq 0$ (vì $h_i + h_j = He$), tức là $h_i \neq h_j$, bởi vì $-h_i = h_j \pmod 2$. Nói cách khác, H là ma trận kiểm tra của mã nhị phân tuyến tính sửa được lỗi đơn (single-error-correcting linear binary code, viết tắt: s.e.c) thì không có cột nào của H là 0 và không có hai cột của H bằng nhau.

Ví dụ 4.7 (Mã sửa được lỗi đơn)

(a) Ma trận kiểm tra H trong (4.11) có các cột thứ hai và thứ tư bằng nhau, và do đó không phải là ma trận kiểm tra của một mã s.e.c. Điều này phù hợp với các kết quả trước đã chỉ ra rằng khoảng cách tối thiểu cho mã này là 2.

(b) Nếu mã chỉ có hai bit kiểm tra, thì ma trận kiểm tra thỏa mãn các điều kiện sửa lỗi đơn chỉ có thể là $H = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$, bởi vì ba cột trên cho thấy chỉ có cột khác không là những cột với hai phần tử. Như đã nhận xét ở trên, hoán vị các cột của H (ví dụ, để sinh ra ma trận H trong (4.9)) không sinh một mã khác.

(c) Khi có ba bit kiểm tra, chỉ có duy nhất một lựa chọn để các cột của ma trận A khác với các cột của I_3 là ma trận đã được chỉ ra trong (4.20). Ma trận H trong (4.20) khi ấy là ma trận kiểm tra cho một mã s.e.c. chiều dài 7. Tương tự, chọn bất kỳ một, hai hoặc ba trong số các cột của A trong (4.20) và phụ thêm chúng vào các cột của I_3, kết quả trong ma trận kiểm tra s.e.c. cho mã chiều dài 4, 5 hoặc 6 tương ứng. Thí dụ, mỗi ma trận

$$H_1 = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}, \qquad H_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{4.21}$$

sinh ra một mã s.e.c. chiều dài 5 với ba bit kiểm tra và hai bit thông tin. Chú ý rằng trong mỗi trường hợp ba cột đầu tiên (có nghĩa là, I_3) được giữ lại.

(d) Ma trận kiểm tra $H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$ có các cột thứ hai và thứ sáu bằng nhau. Vì $x = 010001$ với chữ số 1 trong các vị trí thứ hai và thứ sáu thỏa mãn $Hx = 0$ nên x là một từ mã có trọng số 2. Do đó mã tương ứng với H có khoảng cách tối thiểu 2, vì vậy không sửa được lỗi đơn. Vì H không có cột chỉ gồm toàn các chữ số 0 nên mã không phát hiện được những lỗi đơn.

Phần (c) của ví dụ 4.7 đã chỉ ra rằng một mã s.e.c. với ba bit kiểm tra có chiều dài tối đa là 7. Nói chung, cho một mã chiều dài n với m bit kiểm tra, ma trận kiểm tra sẽ là $H = \underset{m\text{cột}}{\left[I_m \right.} \quad \underset{n-m\ \text{cột}}{\left. A \right]}$ m hàng.

Bởi vì mỗi phần tử trong m phần tử của một cột của A có thể là 0 hoặc 1, nên có nhiều nhất là 2^m cột khác nhau cho lựa chọn A. Nhưng để mã có tính chất s.e.c., cột 0 và m cột của I_m phải bị loại ra từ A. Do đó tổng số các cột có thể chọn chỉ là $2^m - m - 1$. Nghĩa là, đối với một mã nhị phân s.e.c, số $(n - m)$ cột của A phải thỏa mãn $n - m \le 2^m - m - 1$.

Do đó chiều dài của các mã này thỏa mãn các điều kiện $n \le 2^m - 1$. Khi $n = 2^m - 1$ thì mã được gọi là **hoàn hảo** (perfect). Ví dụ, khi có bốn bit kiểm tra, mã s.e.c hoàn hảo có chiều dài 2^4-1 = 15, và số bit thông tin là 15-4=11. Các mã s.e.c không hoàn hảo với bốn bit kiểm tra có ít hơn 11 bit thông tin.

4.3. Thuật toán hội chứng giải mã cho các mã nhị phân

Từ trước đến nay ta đã giải mã bằng cách tìm khoảng cách Hamming giữa một từ nhận được *r* và mỗi từ mã. Khi ấy nguyên tắc lân cận gần nhất chọn từ mã gần nhất với *r* để giả định là từ mã đã được truyền đi. Sơ đồ giải mã này không thực tế khi có một số lượng lớn từ mã. Thay vào đó, cách tiếp cận trực tiếp sử dụng ma trận kiểm tra *H* là thích hợp hơn. Tuy nhiên, nguyên lý "lân cận cực đại (maximum likelihood) vẫn có hiệu lực – Nguyên tắc này giả thiết từ nhận được không có lỗi là gần hơn tất cả các từ khác và từ nhận được với một lỗi là gần hơn từ có hai lỗi hoặc hơn, v.v,...

Nếu *r* là một từ mã, theo định nghĩa nó thỏa mãn *Hr* = 0. Nếu khi truyền tin xảy ra một lỗi đơn ở vị trí thứ *i* thì $r = c + e$, trong đó *e* là một từ có $e_i = 1$, và $e_i = 0$ với mọi *i* khác. Trong trường hợp này một lần nữa $Hc = 0$, nhưng do $He = h_i$ là cột thứ *i* của *H* nên

$$Hr = H(c + e) = Hc + He = 0 + h_i = h_i.$$

Điều này chỉ ra rằng nếu một lỗi đơn xuất hiện trong truyền tin thì *Hr* bằng một cột của *H*. Vì vai trò của nó trong việc xác định từ mã đã được truyền đi, các véc tơ cột *Hr* (nhận được bằng cách nhân ma trận kiểm tra và véctơ cột *r* tạo thành từ từ nhận được *r*) được gọi là **hội chứng** (syndrome) của *r*. Tên gọi này xuất phát từ việc sử dụng thuật ngữ y khoa với nghĩa là "triệu chứng" (symptom).

Để giải mã một từ nhị phân nhận được *r*, ta sử dụng thuật toán sau đây.

Thuật toán hội chứng giải mã cho các mã nhị phân

Bước 1: Tính hội chứng $s = Hr \pmod 2$.

Bước 2: Nếu $s = 0$ thì coi r là một từ mã và lỗi truyền không xảy ra.

Bước 3: Nếu $s =$ cột thứ i của H, giả sử một lỗi đơn đã xảy ra trong bit thứ i, và từ mã đã truyền đi là $r + e$ đã nhận được bằng cách sửa r_i thành $r_i + 1$.

Bước 4: Nếu $s \neq 0$, s khác cột thứ i của H, thì giả thiết rằng nhiều hơn một lỗi đã xảy ra, và đòi hỏi truyền lại tin nhắn.

Ví dụ 4.8 (Ứng dụng của thuật toán)

Xét mã s.e.c. chiều dài 5 với ma trận kiểm tra là ma trận H_1 trong (4.21).

(a) Nếu từ nhận được là r = 11001 thì hội chứng $s = H_1 r$ là

$$s = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.1 + 0.1 + 0.1 + 1.0 + 1.1 \\ 0.1 + 1.1 + 0.0 + 1.0 + 1.1 \\ 0.1 + 0.1 + 1.0 + 0.0 + 1.1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

So sánh s với H_1 trong (4.21) cho thấy rằng nó bằng với cột thứ ba của H_1. Do đó theo bước 3, một lỗi truyền xảy ra trong bit thứ ba, do đó e = 00100. Bit thứ ba của r được sửa thành 0 + 1 = 1, do đó, các từ mã đã truyền là $c = 11101$. Vì $H_1 c = 0$ nên c thực sự là một từ mã.

(b) Nếu từ nhận được là r = 10100 thì hội chứng là:

$$s = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

Rõ ràng s không phải là cột của H_1, do đó, theo bước 4, có nhiều hơn một lỗi đã xảy ra. Ta có thể kiểm tra trong trường hợp này các từ mã đã được truyền đi có thể là 00000 với những lỗi trong bit đầu tiên và thứ ba; hoặc

11101 với những lỗi trong các bit thứ hai và bit thứ năm; hoặc 00111 với các lỗi trong bit thứ nhất, thứ tư và thứ năm. Mã không thể sửa các lỗi bội này.

Phần (b) của ví dụ 4.7 chỉ ra rằng có thể nhận được một hội chứng không phải là một cột của ma trận kiểm tra. Điều này có nghĩa rằng không có từ mã nào mà khoảng cách Hamming đến từ nhận được là 1. Tuy nhiên, nếu mã là hoàn hảo thì điều này không thể xảy ra – một hội chứng sẽ bằng một cột của *H*. Nói cách khác, mỗi từ nhận được có thể được coi là nhận được từ một từ mã duy nhất với nhiều nhất là một lỗi. Để giải thích tại sao, ta nhớ rằng một mã hoàn hảo với m bit kiểm tra có độ dài $n = n = 2^m - 1$, và chứa 2^k từ mã, với $k = n - m$. Có chính xác n từ mà khoảng cách Hamming là 1 đến từ mã 00…0, là các từ mã 10...0, 010...0, 0...010...0, 0...01 (sai khác với từ mã 00…0 duy nhất 1 bit). Tương tự áp dụng cho mỗi từ mã khác. Như vậy tổng số của các từ có chiều dài n mà khoảng cách bằng 1 đến một từ mã là:

$$2^k.n = 2^k\left(2^m - 1\right) = 2^n - 2^k.$$

Bởi vậy, $2^n = 2^k + 2^k.n$, trong đó 2^n là tổng số những từ có độ dài n; 2^k là số những từ mã; $2^k.n$ số những từ có khoảng cách bằng 1 đến từ mã.

Với một mã hoàn hảo, một từ có chiều dài n hoặc chính nó cũng là một từ mã, hoặc có khoảng cách 1 đến một từ mã. Trong trường hợp này Bước 4 của thuật toán không áp dụng được, điều này giải thích tại sao các mã như thế được gọi là “hoàn hảo”. Tất cả điều này giả định rằng tối đa là một lỗi truyền xảy ra. Ví dụ, có thể đối với một từ mã được truyền đi, và nhận được một từ mã khác bởi một số lỗi xảy ra. Thuật toán hội chứng giải mã chỉ có thể kết luận "không có lỗi” trong trường hợp này.

4.4. Mã nhị phân Hamming

Một bất lợi của thuật toán trong phần trước là một hội chứng phải được so sánh với các cột của ma trận kiểm tra. Vào năm 1950, Hamming đã nghĩ ra

một lược đồ, theo đó vị trí của các lỗi đơn có thể nhận được trực tiếp từ hội chứng. Nhớ lại rằng ma trận kiểm tra cho một mã nhị phân s.e.c phải không chứa một cột gồm toàn chữ số 0, hoặc hai cột giống hệt nhau. Hamming chỉ ra rằng, con đường tự nhiên đáp ứng điều kiện này là chọn biểu diễn nhị phân của các số nguyên 1, 2, 3, 4,... như là các cột *H*.

Ví dụ 4.9 Mã Hamming chiều dài 5

Biểu diễn nhị phân của những số nguyên từ đến 5, sử dụng ba bit là:

Số thập phân	**1**	**2**	**3**	**4**	**5**
Nhị phân	001	010	011	100	101

Những số nhị phân này loại trừ 000 và tất cả đều khác nhau, do đó, viết chúng như các cột của một ma trận kiểm tra, mà thực chất sẽ tạo ra một mã s.e.c. có ba bit kiểm tra.

Như trước đây, những bit được đọc từ trái sang phải, và được xếp từ trên xuống dưới, do đó, ví dụ, 001 trở thành $\begin{bmatrix}0\\0\\1\end{bmatrix}$. Thực hiện điều này cho mỗi số nhị phân dẫn đến ma trận:

$$H = \begin{bmatrix}0 & 0 & 0 & 1 & 1\\ 0 & 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 0 & 1\end{bmatrix}. \qquad (4.22)$$

Đây là ma trận kiểm tra cho một mã Hamming nhị phân [5,2]. Chú ý rằng *H* trong (4.22) không còn có dạng mà ba cột đầu tạo nên ma trận I_3 có số chiều 3x3, như trong (4.20). Thay vào đó, chúng là những cột thứ nhất, thứ hai và thứ tư trong (4.20) và chỉ chứa một số 1, tương ứng với biểu diễn nhị phân của 1, 2 và 4. Các bit kiểm tra là x_1, x_2 và x_4. Viết lại các phương trình kiểm tra $Hx = 0$, trong đó $x = x_1x_2x_3x_4x_5$, cho:

$$x_4 + x_5 = 0,\ x_2 + x_3 = 0,\ \ x_1 + x_3 + x_5 = 0.$$

Do $x_i = -x_i$ trong hệ cơ số 2 nên hệ trên có thể được sắp xếp lại như sau:

$$x_1 = x_3 + x_5,\ x_2 = x_3,\ \ x_4 = x_5.$$

Chọn tất cả các giá trị có thể cho các bit thông tin x3 và x5, ta được bảng sau:

Bit kiểm tra			**Bit thông tin**	
x_1	x_2	x_4	x_3	x_5
0	0	0	0	0
1	1	0	1	0
1	0	1	0	1
0	1	1	1	1

Viết các bit theo thứ tự $x_1x_2x_3x_4x_5$ sẽ được bốn từ mã 00000, 11100, 10011, 01111.

Bây giờ xét giải mã bằng cách sử dụng thuật toán hội chứng trong Mục 4.3. Nếu một lỗi đơn xảy ra thì hội chứng bằng một cột của H. Tuy nhiên, không cần phải kiểm tra H trong (4.22) để xem nó là cột nào. Ví dụ, giả sử một từ đã nhận được là r = 01101, khi ấy hội chứng là:

$$s = Hr = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Chuyển véctơ cột s thành số nhị phân $s = (100)_2$, như vậy số này trong hệ thập phân là 4, do đó lỗi ở bit thứ tư, và từ mã được truyền là $r + 00010$ = 01111. Xóa các bit kiểm tra x_1, x_2 và x_4, tin nhắn được giải mã là 11. Điều này minh họa tính chất cơ bản của các mã nhị phân Hamming: biểu diễn nhị phân của hội chứng ngay lập tức chỉ ra bit lỗi.

Nếu có nhiều hơn một lỗi xuất hiện thì Bước 4 của thuật toán hội chứng giải mã khẳng định rằng hội chứng này không phải là cột của ma trận kiểm tra. Ví dụ, nếu $r = 11010$ thì $s = Hr = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$. Chuyển sang số nhị phân $(111)_2$ = 7. Mã có chiều dài chỉ là 5, do đó không thể có lỗi ở bit thứ bẩy: kết luận là nhiều hơn một lỗi đã xảy ra.

Ngẫu nhiên, mã trong ví dụ này có thể được sử dụng để gửi bốn tin nhắn là 00, 10, 01, 11 và là cải tiến của mã lặp trong Ví dụ 2.1 vì ở đây chỉ yêu cầu năm bit để sửa các lỗi đơn, trong khi đó mã lặp trong Ví dụ 2.1 yêu cầu sáu.

4.5. Các tính chất của mã nhị phân Hamming [n,k]

(1) Mã có độ dài n, kích thước k và $m = n - k$ bit kiểm tra. Nếu $n = 2^m - 1$ thì mã là hoàn hảo, nếu $n < 2^m - 1$ mã là **rút ngắn** (shortened).

(2) Với $i = 1,2, ..., n$, cột thứ i của ma trận kiểm tra H kích thước $m \times n$ là vectơ cột được tạo thành từ biểu diễn nhị phân của i sử dụng m bit.

(3) Các bit kiểm tra ở những vị trí, nơi mà các cột của H chỉ chứa một chữ số 1, cụ thể là $x_1, x_2, x_4, x_8, ..., x_p$ với $p = 2^{m-1}$.

(4) Theo cách xây dựng, mã sửa được mọi lỗi đơn, và do đó có khoảng cách tối thiểu là 3. Thật ra, δ thực sự là bằng 3.

(5) Để giải mã một từ nhận được r, ta tính các hội chứng $s = Hr$. Nếu $s = 0$ thì coi từ mã r đã được truyền đi. Ngược lại, nếu $(s)_2 = (j)_{10} \le n$ thì giả sử một lỗi đơn xảy ra trong bit thứ j; nếu $j > n$ thì có nhiều hơn một lỗi xảy ra.

6) Đối với mã Hamming hoàn hảo, mỗi hội chứng (ngoại trừ 0) xuất hiện như là một cột của ma trận kiểm tra, và do đó tương ứng với một lỗi đơn sửa được. Đối với mã Hamming rút gọn, một số hội chứng chỉ ra các lỗi kép.

Ví dụ 4.10 (Mã Hamming hoàn hảo [7, 4])

Ở đây số bit kiểm tra m = 7- 4 = 3, và $n = 2^3 - 1$. Biểu diễn nhị phân $6=(110)_2$ và $7 = (111)_2$ cho hai cột bổ sung sẽ được nối thêm vào (4.22). Do đó ma trận kiểm tra 3x7 là:

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \qquad (4.23)$$

Ví dụ, giả sử một từ đã nhận được là r = 0110111, và tối đa một lỗi xảy ra trong truyền tin. Hội chứng là:

$$s = Hr = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

Điều này dẫn đến $(101)_2 = 5$ chỉ ra rằng lỗi xuất hiện tại bit thứ 5 và khi ấy từ mã đã được chuyển là 0110011. Loại các bit kiểm tra x_1, x_2, x_4 dẫn tới tin nhắn được giải mã là 1011.

4.6. Các *p*-mã Hamming

Một p–mã C có độ dài n đã được định nghĩa trong §3 là một tập hợp các từ mã $x = x_1x_2...x_n$ với $x_i \in \mathbb{Z}_p$. Dưới đây ta chỉ xem xét các mã với p là một số nguyên tố, do đó $\mathbb{Z}_p$ là một trường hữu hạn. Định lý 3.1 về phát hiện và sửa lỗi trong §3 được áp dụng cho bất kỳ p–mã nào, nhưng định nghĩa tuyến tính trong mục 4.1 cần sửa đổi theo cách sau.

Cho hai p–từ mã x và $y = y_1y_2...y_n$, **tổng** $z=x+y$ có các chữ số $z_i = (x_i + y_i)(mod\ p)$, i = 1, 2, …, n. Khi ấy C là **tuyến tính** khi và chỉ khi

(i) $x + y \in C$, với mọi $x, y \in C$ và

(ii) $ax \in C$ với mọi $a \in \mathbb{Z}_p$.

Đối với mã nhị phân, điều kiện (ii) trở thành luôn đúng vì $\mathbb{Z}_2 = \{0,1\}$.

Trọng số của p–từ mã bây giờ là số các chữ số khác không của nó.

Với định nghĩa mở rộng này, Định lí quan trọng (4.1) vẫn còn đúng, tức là khoảng cách tối thiểu của một mã tuyến tính bằng trọng số nhỏ nhất của từ mã khác không.

Mã tuyến tính có khả năng sửa mọi lỗi đơn ít nhất phải có khoảng cách tối thiểu là 3. Điều này trở về yêu cầu đầu tiên là ma trận H phải không có cột gồm toàn các số 0, giống như trong mã nhị phân; yêu cầu thứ hai trước kia trong mã nhị phân là không có hai cột của ma trận kiểm tra bằng nhau, bây giờ được thay thế bởi điều kiện là không có cột nào là *bội* khác 0 của cột khác; bội này được xác định bằng cách nhân mỗi phần tử của véctơ x với $c \in \mathbb{Z}_p$, đó là $c\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} cx_1 \\ cx_2 \\ \cdot \\ \cdot \end{bmatrix} (\text{mod } p)$.

Phương trình kiểm tra (4.10) bây giờ trở thành $Hx \equiv 0 \pmod{p}$.

Ví dụ 4.11 Mã tuyến tính tam phân

Ma trận kiểm tra $\begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \end{bmatrix}$ không phải là ma trận cho một mã tam phân s.e.c., bởi vì các cột thứ tư là bằng 2 lần cột thứ ba, đó là,

$$2\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} (\text{mod } 3). \qquad (4.24)$$

Tuy nhiên, ma trận kiểm tra

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 \end{bmatrix} \qquad (4.25)$$

xác định một mã tam phân tuyến tính s.e.c, bởi vì không có cột nào là bội của các cột khác. Để xác minh điều này, ta chỉ cần viết ra tất cả các bội khác không của các cột và xác nhận rằng không có cột nào lặp lại. Đối với mã này, từ mã $x = x_1x_2x_3x_4$ được bằng cách giải các phương trình kiểm tra $Hx \equiv 0 (\text{mod } 3)$ với H trong (4.25). Điều này cho

$$x_2 + x_3 + x_4 = 0; \qquad x_1 + x_3 + 2x_4 = 0.$$

Lấy x_3 và x_4 là chữ số thông tin, những chữ số kiểm tra có thể được biểu diễn như sau (vì -1 = 2, - 2 = 1 theo môđun 3):

$$\begin{cases} x_1 = -x_3 - 2x_4 = 2x_3 + x_4; \\ x_2 = -x_3 - x_4 = 2x_3 + 2x_4. \end{cases} \tag{4.26}$$

Các chữ số x_3 và x_4 độc lập có thể lấy giá trị 0, 1, 2, vậy có $3^2 = 9$ từ mã. Như đối với các mã nhị phân tuyến tính, các giá trị của các chữ số kiểm tra nhận được từ (4.26), theo môđun 3. Ví dụ, khi $x_3 = 1$, $x_4 = 2$ thì x_1=2 +2 =1, $x_2 = 2 + 4 = 0$, và các từ mã tương ứng là 1012. Một số từ mã khác được cho trong bảng sau:

Chữ số kiểm tra		**Chữ số thông tin**	
x_1	x_2	x_3	x_4
1	2	0	1
2	1	0	2
2	2	1	0
1	0	1	2

Các mã được xác định bởi ma trận kiểm tra H trong (4.25) thực sự là một mã tam phân Hamming chiều dài 4 với hai bit kiểm tra và hai bit thông tin. Ghi các cột của H như những số trong hệ cơ số 3:

$$(01)_3 = 1 \qquad (10)_3 = 3, \qquad (11)_3 = 4, \qquad (12)_3 = 5, \tag{4.27}$$

ta thấy các cột của H thỏa mãn các điều kiện quy định cho một mã s.e.c.

4.7. Các tính chất của *p*-mã Hamming [n,k]

(1) Mã có chiều dài $n=\left(p^m-1\right)/\left(p-1\right)$, m chữ số kiểm tra, kích thước $k=n-m$, và chứa p^k từ mã.

(2) Các cột của ma trận kiểm tra H kích thước $m\times n$ là các véc tơ cột tạo bởi các số cơ sở p với m chữ số đầu tiên, trong đó chữ số đầu tiên khác không là 1. Những số này được chọn theo chiều tăng của biên độ, nghĩa là:

00..01, 00…010, 00..011, 00…012, …, 00…01(p - 1), 00…0100,… (4.28)

(3) Các chữ số kiểm tra ở tại các vị trí mà tại đó những cột của H có các số khác 0 đơn bằng 1.

(4) Mã có khoảng cách tối thiểu 3, sửa được mọi lỗi đơn, và là hoàn hảo.

(5) Để giải mã một p-từ $r=r_1r_2...r_n$ nhận được, ta tính các hội chứng $s=Hr\left(\mathrm{mod}\,p\right)$. Nếu s = 0 thì coi từ mã r đã được truyền đi. Ngược lại, thì $s=eh_i$, trong đó $e\neq 0$ và $e\in\mathbb{Z}_p$, và h_i là cột thứ i của H. Một lỗi đơn biên độ e được coi là tại chữ số thứ i, và chữ số được sửa là r_i-e.

Khẳng định mã là hoàn hảo trong (4) cần được chứng minh. Ta phải chỉ ra rằng mỗi hội chứng khác 0 là bằng e lần một cột nào đó của H, trong đó $e\neq 0$ và $e\in\mathbb{Z}_p$.

Bất kì hội chứng khác không s có m phần tử, và có thể được viết trong dạng chuyển vị

$$[0, 0, \ldots, 0, s_1, s_2, \ldots, s_{m-q}], \tag{4.29}$$

có q số 0, trong đó $0\le q<m$ và s_1 là số đầu tiên khác 0 của s, với mọi $s_i\in\mathbb{Z}_p$. Để $s=e$ lần một cột của H thì cột của H phải có dạng:

$$\left[0, 0, ..., 0, 1, c_2, c_3, ..., c_{m-p}\right]. \tag{4.30}$$

So sánh mỗi phần tử trong (4.29) với e lần phần tử tương ứng trong (4.30) cho:

$$s_1 = e, \qquad s_2 = ec_2, \qquad s_3 = ec_3, \quad \ldots, \quad s_{m-q} = ec_{m-q}. \tag{4.31}$$

Khi p là số nguyên tố, mỗi phần tử của $\mathbb{Z}_p$ đều có số nghịch đảo. Nói riêng, $e^{-1} = s_1^{-1} \in \mathbb{Z}_p$, do đó (4.32) có thể được giải được:

$$c_2 = s_1^{-1}s_2,\ c_3 = s_1^{-1}s_3,\ \ldots,\ c_{m-q} = s_1^{-1}s_{m-q}.$$

Điều này chỉ ra rằng, (4.30) thực sự biểu diễn cột của H hay mã là hoàn hảo.

Ví dụ 4.12 Các *p*-mã Hamming

a) Khi $p = 5$ và $m = 3$, chiều dài của mã được cho bởi tính chất (1) là: $n = (5^3 - 1)/(5-1)$ và $k = 31 - 3 = 28$.

Các số với ba chữ số trong cơ số 5 nhận được từ (4.28) là:

001, 010, 011, 012, 013, 014, 100, 101, 102, 103, 104, 110, 111, 112, 113, 114,120,121,122,123, 124, 130, 131, 132, 133, 134, 140, 141, 142, 143, 144.

Ta có thể kiểm tra rằng trong hệ thập phân 31 số trên tương đương với các số 1, 2, 7, 9, 25, 26, …, 48, 49, xác nhận rằng những số này đã được sắp xếp theo thứ tự tăng dần.

Chúng được sử dụng như các cột của ma trận kiểm tra 3x31 tương ứng với 001, 010, và 100, và có $5^{28} \approx 3.73 \times 10^{19}$ từ mã.

b) Khi $p = 3$, $m = 2$ mã có chiều dài $n = (3^2 - 1)/(3-1) = 4$. Các số (4.28) được liệt kê trong (4.27), và ma trận kiểm tra H là (4.25).

Giả sử một từ nhận được là $r = 1200$. Theo (4.24) ta có

$$s = H\begin{bmatrix}1\\2\\0\\0\end{bmatrix} = \begin{bmatrix}0 & 1 & 1 & 1\\ 1 & 0 & 1 & 2\end{bmatrix}\begin{bmatrix}1\\2\\0\\0\end{bmatrix} = \begin{bmatrix}0.1+1.2+1.0+1.0\\ 1.1+0.2+1.0+2.0\end{bmatrix} = \begin{bmatrix}2\\1\end{bmatrix} = 2\begin{bmatrix}1\\2\end{bmatrix} (\text{mod } 5).$$

Hội chứng này gấp 2 lần cột thứ tư của H, bởi vậy, theo tính chất (5) thì có một lỗi $e = 2$ ở chữ số thứ tư. Chữ số cần sửa của từ nhận được là $r_4 - 2 = 0 - 2 = 1 (\text{mod} 3)$, do đó từ mã được truyền đi là 1201.

c) Khi p = 5 và m = 2 chiều dài của mã là $n=\frac{(5^2-1)}{(5-1)}=6$ và $k=6-2=4$.

Những số cơ số 5 trong (4.28) với hai chữ số theo thứ tự tăng dần là: 01, 10, 11, 12, 13, 14. Chuyển những số này thành những cột theo cách thông thường tạo ra ma trận kiểm tra

$$H=\begin{bmatrix}0 & 1 & 1 & 1 & 1 & 1\\ 1 & 0 & 1 & 2 & 3 & 4\end{bmatrix}. \qquad (4.32)$$

Giả sử từ đã nhận được là $r=202123$. Sử dụng H trong (4.32), hội chứng là:

$$s=H\begin{bmatrix}2\\0\\2\\1\\2\\3\end{bmatrix}=\begin{bmatrix}0 & 1 & 1 & 1 & 1 & 1\\ 1 & 0 & 1 & 2 & 3 & 4\end{bmatrix}\begin{bmatrix}2\\0\\2\\1\\2\\3\end{bmatrix}=\begin{bmatrix}0.2+1.0+1.2+1.1+1.2+1.3\\ 1.2+0.0+1.2+2.1+3.2+4.3\end{bmatrix}=\begin{bmatrix}3\\4\end{bmatrix}(\text{mod}5).$$

Dễ dàng thấy rằng $s=3\begin{bmatrix}1\\3\end{bmatrix}=3h_5$. Vì vậy, có một lỗi e = 3 trong chữ số thứ năm của r, được sửa thành 2 – 3 = –1= 4, từ mã được truyền đi là 202143.

d) Khi p =3, m = 3 mã tam phân Hamming có chiều dài $(3^3-1)/(3-1)=13$, kích thước k = 13 – 3 = 10. Đổi các số trong (4.28) để nhận được các cột của ma trận kiểm tra

$$H=\begin{bmatrix}0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2\\ 1 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2\end{bmatrix}.$$

Các chữ số kiểm tra là x_1 , x_2 và x_5, và có $3^{10}=59.049$ từ mã. Nếu một từ r_1=1102112100112 nhận được thì ta có thể kiểm tra rằng $s = Hr_1 = 0$, do

đó, các từ mã r_1 đã được truyền đi. Nếu một từ khác r_2 = 1000101220120 nhận được, ta có thể tính $s = Hr_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = h_3$.

Chữ số thứ ba của r_2 được sửa chữa là $0-1=2$, từ mã được truyền đi là 1020101220120.

§5. Mã thập phân

Một số mã thập phân đã được gặp trên số hiệu hàng hóa châu Âu (EAN) (ví dụ 2.4), ngân phiếu Mỹ (ví dụ 2.5), mã sách tiêu chuẩn quốc tế (ISBN). Điểm đặc trưng chung của chúng là các chữ số của từ mã thuộc $\{1,2,...,9\}$ (ngoại trừ các chữ số kiểm tra), mặc dù số học modula rất khác nhau giữa đồng dư 9 và 10 hoặc 11 tùy theo những mã đặc biệt.

5.1. Mã số sách tiêu chuẩn quốc tế (ISBN)

Mỗi cuốn sách hoặc văn hóa phẩm xuất bản được đồng nhất bởi ISBN của nó, thí dụ như trong Hình 1.2, ISBN là một mã từ gồm 10 chữ số $x_1x_2...x_{10}$, trong đó x_1 đến x_9 là các chữ số thập phân, nhưng chữ số kiểm tra x_{10} có thể nhận thêm giá trị x_{10}, được ký hiệu là X (chữ số La Mã có giá trị là 10). Điều này là bởi vì ISBN được xác định với số học môđun 11, để mã được định nghĩa trên trường hữu hạn $\square_{11}$. Hàng số ở dưới thấp hơn trong Hình 1.2 là số hiệu hàng hóa châu Âu (EAN) cho sách. Những chữ số đầu tiên của một ISBN, được gọi là 'nhóm định danh' (Group Identifier), ký hiệu quốc gia, hoặc nhóm các quốc gia. Ví dụ, $x_1 = 0$ được sử dụng cho tất cả các sách (cho dù là bằng tiếng Anh hoặc không) được xuất bản tại Hoa Kỳ, Vương quốc Anh, Canada. Australia và một số nước khác; x_1 = 3 chỉ ra một cuốn sách được xuất bản bằng tiếng Đức; Đan Mạch có x_1x_2= 87, và Thụy Điển x_1x_2=90. Phần thứ hai của ISBN, được gọi là "nhà xuất bản tiền tố" (Publisher Prefix)

có thể bao gồm hai, ba, bốn, năm, sáu hoặc bảy chữ số. Những chữ này đồng nhất với nhà xuất bản, ví dụ như những chữ số x_2x_3=13, là kí hiệu của Prentice Hall. Phần thứ ba của ISBN có thể dài từ một đến sáu chữ số và là "Số tiêu đề" (Title Number), đó là số được gán cho những cuốn sách cụ thể của nhà xuất bản, ví dụ, 053101 trong Hình 1.2. Chiều dài của số tiêu đề phụ thuộc vào độ dài của các phần trước của ISBN, nhưng Identifier Group, Publisher Prefix và Title Number luôn có tổng là chín chữ số. Chữ số cuối x_{10} là chữ số kiểm tra, và được chọn sao cho tổng kiểm tra

$$S = \sum_{i=1}^{10} ix_i = x_1 + 2x_2 + 3x_3 + ... + 10x_{10} \tag{5.1}$$

là một bội số của 11, có nghĩa là,

$$S \equiv 0 \pmod{11}. \tag{5.2}$$

Những dấu gạch nối thường được chèn vào giữa những phần khác nhau của ISBN, nhưng không có ý nghĩa toán học.

Đặt tổng (5.1) bằng không, và sử dụng $-10 \equiv 1 \pmod{11}$ ta đi đến biểu diễn

$$x_{10} = \sum_{i=1}^{9} ix_i \pmod{11}. \tag{5.3}$$

Điều này cho phép tính chữ số kiểm tra theo chín chữ số đầu tiên của ISBN. Dễ dàng thấy rằng nếu tổng bên phải của (5.3) là số thập phân có ba chữ số abc thì (5.3) được rút gọn thành

$$x_{10} = (a - b + c) \pmod{11}. \tag{5.4}$$

Tương tự, nếu trong (5.1) $S=(s_1s_2s_3)_{10}$, thì (5.2) tương đương với $s_1 - s_2 + s_3 \equiv 0 \pmod{11}$. Tất nhiên, dễ dàng để tính toán tổng trong (5.1) và (5.3) bằng cách sử dụng máy tính. Tuy nhiên, ISBN được đưa vào sử dụng khoảng năm 1968, trước khi có các máy tính giá rẻ, do đó, người ta đã tạo ra một bảng sử dụng rất đơn giản cho các thủ thư. Bảng này chỉ cần làm các phép cộng mà không cần phép nhân nào. Điều kiện (5.2) được kiểm tra những quy tắc dưới đây:

(1) Xây dựng một mảng với ba hàng và mười cột. Các mục trong dòng đầu tiên là $x_1, x_2, ..., x_{10}$, và trong cột đầu tiên là x_1, x_1, x_1.

(2) Nếu các số trong một cột là a_1, a_2, a_3 thì các số b_2, b_3 trong cột bên cạnh bên phải được tính theo các mũi tên sau.

Hàng đầu $a_1 \quad b_1$

$$a_2 \longrightarrow b_2 = a_2 + b_1$$

$$a_3 \longrightarrow b_3 = a_3 + b_2$$

(3) Điều kiện (5.2) tương đương với chữ số cuối cùng trong dòng dưới cùng của bảng là 0(mod 11).

Để kiểm tra (3), ta xây dựng bảng như sau:

$$\begin{array}{ccccccc} x_1 & & x_2 & & x_3 & \quad x_4 \quad & \ldots \quad x_{10} \\ x_1 & \longrightarrow & (x_1 + x_2) & \longrightarrow & (x_1 + x_2 + x_3) & (x_1 + x_2 + x_3 + x_4) & \ldots \quad T_1 \\ x_1 & \longrightarrow & (2x_1 + x_2) & \longrightarrow & (3x_1 + 2x_2 + x_3) & (4x_1 + 3x_2 + 2x_3 + x_4) & \ldots \quad T_2 \end{array}$$

Ta có thể kiểm tra rằng $T_1 = x_1 + x_2 + x_3 + ... + x_{10}$, và số cuối cùng trong dòng dưới cùng là $T_2 = 10x_1 + 9x_2 + 8x_3 + ... + 2x_9 + x_{10}$.

Hơn nữa, ta dễ dàng thấy rằng $T_2 + S = 11T_1$, trong đó S là tổng trong (5.2), do đó, $T_2 \equiv 0 (\text{mod } 11)$ khi và chỉ khi $S \equiv 0 (\text{mod } 11)$.

Việc áp dụng các quy tắc (2) có thể làm đơn giản bằng cách thực hiện từng phép cộng riêng lẻ theo môđun 11 trong quá trình xây dựng mảng.

Ví dụ 5.1 (Kiểm tra một ISBN)

Xét ISBN 3880531013 như trong Hình 1.2. Tổng kiểm tra S trong (5.2) là:

$$\begin{aligned} 3880531013 &= 1\times 3 + 2\times 8 + 3\times 8 + 4\times 0 \\ &+ 5\times 5 + 6\times 3 + 7\times 1 + 8\times 0 + 9\times 1 + 10\times 3 \\ &\equiv 3 + 5 + 2 + 3 + 7 + 7 + 9 + 8 (\text{mod } 11) \equiv 0 (\text{mod } 11). \end{aligned}$$

Thỏa mãn điều kiện đúng của ISBN.

Hình 1.2

Sử dụng những quy tắc (1) và (2), mảng được cấu tạo như sau.

3	8	8	0	5	3	1	0	1	3	← ISBN
3	11	19	19	24	27	28	28	29	32	
3	14	33	52	76	103	131	159	188	220	

Số cuối cùng của dòng dưới cùng là 220 đồng dư với 0 (mod 11).

Phiên bản đơn giản hơn của bảng này sử dụng phép cộng theo môđun 11 trong suốt quá trình. Ví dụ, các mục trong cột thứ hai giảm thành $8+3=0$, $3+0=3(\text{mod}11)$. Bảng đơn giản hơn bảng trên sẽ là:

3	8	8	0	5	3	1	0	1	3	← ISBN
3	0	8	8	2	5	6	6	7	10	
3	3	0	8	10	4	10	5	1	11	(5.4)

Và số cuối cùng là 11 đồng dư với 0 (mod 11), đúng như trên.

Qui trình này cũng cho phép ta dễ dàng tìm được chữ số kiểm tra nếu biết các giá trị của x_1 đến x_9. Trong ví dụ này, giả sử x_{10} chưa được biết. Hai cột cuối của mảng sẽ là:

1	x_{10}
7	$x_{10}+7$
1	$x_{10}+8$

Và đòi hỏi $x_{10}+8\equiv 0(\text{mod } 11)$ cho giá trị $x_{10}=3$.

Ví dụ 5.2 (Sửa ISBN)

Giả sử rằng chữ số thứ sáu (bằng 3) trong ISBN ở ví dụ 5.1 đã vô tình bị mất, do đó, ISBN có dạng $38805x1013$. Hoặc là tính S từ (5.2), hoặc sử dụng các mảng đơn giản (5.4). Năm cột đầu tiên (5.4) hoàn toàn như trước, nhưng phần còn lại của mảng đó bây giờ là

5	x	1	0	1	3
2	$x+2$	$x+3$	$x+3$	$x+4$	$x+7$
10	$x+1$	$2x+4$	$3x+7$	$4x$	$5x+7$

trong đó các tổng đã được lấy đồng dư theo mod 11. Yêu cầu số cuối cùng ở dòng dưới là

$$5x+7\equiv 0(\mathrm{mod}11). \tag{5.5}$$

Lời giải là

$$5x\equiv -7\equiv 4(\mathrm{mod}11).$$

Bởi vậy

$$x=5^{-1}\times 4=9\times 4=36\equiv 3.$$

Đó chính là giá trị của các chữ số bị mất. Chú ý rằng do 11 là số nguyên tố nên tồn tại phần tử nghịch đảo trong $\square_{11}$. Ngoài ra, ta có thể giải (5.5) chỉ đơn giản bằng cách thử $x=1, 2, 3$ cho đến khi nhận được giá trị đúng là $x=3$. Phương trình (5.5) trong $\square_{11}$ cho nghiệm duy nhất.

Giả sử nhận được số thập phân gồm 10 chữ số thập phân và phát hiện ra nó không có trong ISBN. Nếu không biết những chữ số nào lỗi thì có thể đã được truyền đi nhiều ISBN. Thậm chí giả thiết thông thường thích hợp nhất là chỉ có một lỗi đơn trong một chữ số, thì cũng không đủ để sửa một lỗi như vậy. Tuy nhiên, mã ISBN phát hiện mọi lỗi trong đó hai chữ số (thông thường, nhưng không phải nhất thiết, liền kề) ngẫu nhiên bị đổi chỗ. Để thấy

điều này, giả sử một ISBN đúng $x_1x_2...x_{10}$ đã được truyền đi, và từ nhận được có x_j và x_k bị đổi chỗ, với $j<k$ và $x_j \neq x_k$ (nếu $x_j = x_k$ thì không có lỗi). Cho từ nhận được ta có tổng kiểm tra *S* trong (5.2) chứa các số hạng $jx_k + kx_j$ thay vì $jx_j + kx_k$. Dễ dàng thấy rằng tổng kiểm tra sai khác là $(x_j - x_k)(k - j)$ và tích này không thể đồng dư với 0 (mod 11) vì mỗi số hạng đều khác không. Do đó tổng kiểm tra cho từ nhận được không đồng dư với 0 (mod 11), vì vậy lỗi được phát hiện. Nếu như biết hai chữ số liền kề đã được đổi chỗ thì lỗi này có thể sửa được.

5.2. Mã sửa lỗi đơn

Mã ISBN hiện nay đã được cải tiến để sửa lỗi đơn. Mã này bao gồm các từ mã có 10 chữ số $x_1, x_2, ..., x_{10}$ thỏa mãn với các phương trình kiểm tra trong (5.1) như mã ISBN, cụ thể là

$$S_1 \equiv \sum_{i=1}^{10} ix_i \equiv 0 \pmod{11} \tag{5.6}$$

bổ xung thêm một phương trình kiểm tra tính chẵn lẻ

$$S_2 \equiv \sum_{i=1}^{10} x_i \equiv 0 \pmod{11}. \tag{5.7}$$

Vì có hai phương trình kiểm tra nên bây giờ có hai chữ số kiểm tra x_9 và x_{10}. Những phương trình kiểm tra này xác định một 11-mã, nhưng bỏ qua mọi từ mã chứa chữ số X (=10), mã kết quả chỉ còn chứa các chữ số thập phân đúng (các chữ số 1, 2, ..., 9).

Giả sử một lỗi đơn *e* xảy ra ở vị trí thứ *i* của một từ mã đã được truyền đi, khi ấy từ nhận được r có $r_i = x_i + e$. Như đối với mã ISBN, với từ *r* nhận được tổng kiểm tra (5.6) có một số hạng bổ xung *ie*, vì vậy

$$S_1 \equiv ie \pmod{11}. \tag{5.8}$$

Tổng kiểm tra thứ hai (5.7) trở thành $S_2 \equiv e \pmod{11}$. Vì thế biên độ của lỗi này là $e \equiv S_2 \pmod{11}$, và từ (5.8) vị trí của lỗi này là:

$$i \equiv S_1 e^{-1} \pmod{11} \equiv S_1 S_2^{-1} \pmod{11}.$$

Thuật toán giải mã cho mã này là như sau, trong đó tất cả các phép toán đã được lấy đồng dư theo mod 11:

(1) Với từ nhận được *r*, tính hội chứng $Hr = \begin{bmatrix} S_2 \\ S_1 \end{bmatrix}$ với

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix}. \qquad (5.9)$$

(2) Nếu $S_1 = 0$, $S_2 = 0$ thì không có lỗi, và từ mã r đã được truyền đi.

(3) Nếu $S_1 \neq 0$ và $S_2 \neq 0$ thì giả thiết một lỗi duy nhất xảy ra tại chữ số $i = S_1 S_2^{-1}$, và chữ số chính xác là $r_i - S_2$.

(4) Nếu $S_1 = 0$ hoặc $S_2 = 0$ (nhưng không cả hai) thì ít nhất hai lỗi đã được phát hiện.

Thật ra (4) luôn xảy ra khi hai chữ số được đổi chỗ, nhưng khác với ISBN, mã này có thể phát hiện tất cả các lỗi xảy ra trên hai chữ số. Điều này là vì mã có khoảng cách tối thiểu là 3. Cách dễ nhất để chứng minh điều này là chú ý rằng các ma trận kiểm tra trong (5.9) chính là ma trận kiểm tra cho 11- mã Hamming

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix},$$

nhưng bỏ đi hai cột đầu tiên. Điều này không làm thay đổi khoảng cách tối thiểu, đều là 3 cho các *p*- mã Hamming.

Ví dụ 5.3 (Các ứng dụng của lược đồ giải mã)

(a) Với một từ được 0206211909 dễ dàng để tính được bằng cách sử dụng (5.6) và (5.7) là

$$S_1 = 213 \equiv 4 \pmod{11}, \qquad S_2 = 30 \equiv 8 \pmod{11}.$$

Do đó theo bước (3) ở trên, một lỗi duy nhất đã xảy ra ở vị trí thứ i, với

$$i = 4 \times 8^{-1} = 4 \times 7 = 28 \equiv 6 \pmod{11}.$$

Vì vậy chữ số thứ sáu của từ nhận được được sửa lại là

$r_6 - 8 = 1 - 8 = -7 \equiv 4 \pmod{11}$, do đó, từ mã được truyền là 0206241909.

(b) Với từ nhận được 5764013052 ta kiểm tra $S_1 \equiv 1$ và $S_2 \equiv 0 \pmod{11}$. Theo bước (4) có ít nhất hai lỗi trong từ nhận được, và yêu cầu truyền tin lại.

5.3. Mã sửa lỗi kép

Các mã được trình bày ở trên có khả năng, trong trường hợp tốt nhất, sửa được các các lỗi đơn. Bây giờ ta có thể mô tả mã sửa được *tất cả* các lỗi kép, nghĩa là, những lỗi trong hai chữ số của một từ mã.

Ta bắt đầu với mã s.e.c và chọn những từ mã thỏa mãn ngoài (5.6) và (5.7) còn thêm hai phương trình kiểm tra

$$S_3 = \sum_{i=1}^{10} i^2 x_i \equiv 0 \pmod{11} \quad \text{và} \quad S_4 = \sum_{i=1}^{10} i^3 x_i \equiv 0 \pmod{11}. \tag{5.10}$$

Bây giờ có bốn chữ số kiểm tra x_7, x_8, x_9, x_{10}. Giả sử một từ mã $x_1 x_2 ... x_{10}$ đã được truyền và hai lỗi xảy ra ở các vị trí thứ i và thứ j với biên độ e_1 và e_2 tương ứng, khi ấy từ r nhận được có $r_i = x_i + e_1$, $r_j = x_j + e_2$. Từ các biểu diễn của tổng kiểm tra suy ra rằng trong trường hợp này:

$$\left.\begin{aligned} S_1 &= ie_1 + je_2, & S_2 &= e_1 + e_2, \\ S_3 &= i^2 e_1 + j^2 e_2, & S_4 &= i^3 e_1 + j^3 e_2. \end{aligned}\right\}. \tag{5.11}$$

Bốn phương trình trong (5.11) sẽ giải được đối với bốn ẩn số i, j, e_1, e_2. Một số tính toán đại số thích hợp dẫn đến kết quả là i và j (các vị trí của lỗi) là hai nghiệm của phương trình bậc hai:

$$a x^2 + bx + c = 0, \tag{5.12}$$

trong đó:

$$a = S_1^2 - S_2 S_3, \qquad b = S_2 S_4 - S_1 S_3, \qquad c = S_3^2 - S_1 S_4. \tag{5.13}$$

Khi i và j đã được tìm thấy, thì rất dễ dàng giải hai phương trình đầu tiên trong (5.11) để được

$$e_2 = \left(iS_2 - S_1\right)\left(i - j\right)^{-1}, \quad e_1 = S_2 - e_2. \tag{5.14}$$

Lưu ý là nếu chỉ một lỗi xảy ra, thí dụ , với $e_1 \neq 0$, $e_2 \neq 0$ thì trong (5.11) $S_1 = ie_1$, $S_2 = e_1$, $S_3 = i^2 e_1$, $S_4 = i^3 e_1$, và thay thế các giá trị này vào (5.13) ta được $a = 0$, $b = 0$, $c = 0$.

Nghiệm của phương trình bậc hai (5.12) có dạng

$$i, j = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{5.15}$$

Các căn bậc hai của phần tử q (nếu chúng tồn tại) trong $\square_{11}$ được cho bởi:

$$\left.\begin{array}{lccccc} q & 1 & 3 & 4 & 5 & 9 \\ \sqrt{q} & 1 \text{ hoặc } 10 & 5 \text{ hoặc } 6 & 2 \text{ hoặc } 9 & 4 \text{ hoặc } 7 & 3 \text{ hoặc } 8 \end{array}\right\}. \tag{5.16}$$

Thuật toán giải mã cho mã này như sau, trong đó tất cả các tính toán số học được lấy theo môđun 11:

(1) Với một từ r nhận được, tính hội chứng $S = H.r = \begin{bmatrix} S_2 \\ S_1 \\ S_3 \\ S_4 \end{bmatrix}$, trong đó

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & \ldots & 1 \\ 1 & 2 & 3 & 4 & \ldots & 10 \\ 1 & 2^2 & 3^2 & 4^2 & \ldots & 10^2 \\ 1 & 2^3 & 3^3 & 4^3 & \ldots & 10^3 \end{bmatrix}. \tag{5.17}$$

(2) Nếu $S = 0$ (có nghĩa là, $S_1 = S_2 = S_3 = S_4 = 0$) thì giả thiết không có lỗi, và từ mã r đã được truyền đi.

(3) Nếu $S \neq 0$ và $a = b = c = 0$, thì có một lỗi đơn xảy ra tại chữ số $i = S_1 S_2^{-1}$, và chữ số được sửa là $r_i - S_2$.

(4) Nếu $S \neq 0$ và $a \neq 0, c \neq 0$, và $q = b^2 - 4ac$ có một giá trị căn bậc hai trong $\square_{11}$ theo (5.16), khi đó giả thiết là có lỗi ở các vị trí thứ i, j cho bởi (5.15). Các chữ số này được sửa là $r_i - e_1, r_j - e_2$, trong đó e_1 và e_2 được cho bởi (5.14).

(5) Nếu các điều kiện (2), (3) hoặc (4) không thỏa mãn, thì có ít nhất ba lỗi đã được phát hiện. Từ (5.16) ta thấy rằng điều này bao gồm cả các trường hợp (4) khi q nhận một trong giá trị 2, 6, 7, 8, bởi vì các số này không có căn bậc hai trong $\square_{11}$.

Có thể chỉ ra rằng mã có khoảng cách tối thiểu là 5, do đó, theo Định lý trong §3, mã này sửa được tất cả các lỗi kép.

Ví dụ 5.4 (Ứng dụng của thuật toán giải mã)

(a) Giả sử một từ được nhận là 3254571396. Các tổng S_1, S_2, S_3 và S_4 trong (5.6), (5.7) và (5.8) được tính toán với số học môđun 11. Để thuận tiện cho việc tính toán ta đưa ra bảng sau:

x_i	**3**	**2**	**5**	**4**	**5**	**7**	**1**	**3**	**9**	**6**
i	1	2	3	4	5	6	7	8	9	10
i^2	1	4	9	5	3	3	5	9	4	1
i^3	1	8	5	9	4	7	2	6	3	10

Trước tiên, $S_2 = \sum x_i = 1$, và lấy tích theo môđun 11 của dòng đầu tiên của bảng với các hàng tiếp theo ta được:

$$S_1 = 1\times 3 + 2\times 2 + 3\times 5 + 4\times 4 + 5\times 5 + 6\times 7 + 7\times 1 + 8\times 3 + 9\times 9 + 10\times 6 = 2$$

Và tương tự

$$S_3 = \sum i^2 x_i = 10; \qquad S_4 = \sum i^3 x_i = 3.$$

Từ (5.13) ta có

$a = 4 - 1\times 10 = -6 = 5;$ $b = 1\times 3 - 2\times 10 = -17 = 5;$ $c = 100 - 2\times 3 = 6.$

Vì vậy, đại lượng $q = b^2 - 4ac$ trong bước (2) là:

$$Q = 25 - 4\times 5\times 6 = -95 = 4.$$

Nghĩa là $\sqrt{q} = 2$ trong (5.16). Do đó các điều kiện trong bước (4) của thuật toán được thỏa mãn. Vậy có hai lỗi đã được truyền đi. Từ (5.15) các lỗi này xảy ra ở các vị trí $i, j = (-5 \pm 2)10^{-1}$. Vì $10^{-1} = 10$ nên

$$i = -3\times 10 = -30 = 3; \quad j = -7\times 10 = 7.$$

Biên độ của các lỗi tính được từ (5.14) là $e_2 = (3\times 1 - 2)(3-7)^{-1} = 7^{-1} = 8$ và $e_1 = 1 - 8 = 4$. Do đó các giá trị chính xác của các chữ số thứ ba và thứ bảy là $r_3 - 4 = 5 - 4 = 1$ và $r_7 - 8 = 1 - 8 = -7 = 4$.

Từ nhận được khi ấy được giải mã là 3214574396.

Nhận xét rằng sử dụng giá trị $\sqrt{4} = 9$ trong (5.16) ta được

$$i, j = (-5 \pm 9)10^{-1} = (-5 \pm 9)10 = -30; 40 = 3; 7$$

như trên khi chọn giá trị $\sqrt{4} = 2$.

Cũng chú ý rằng điều kiện $c \neq 0$ trong Bước (4) là cần thiết cho việc sửa chữa hai lỗi, vì nếu $c = 0$ thì (5.12) có một nghiệm $i = 0$, vô lí.

(b) Giả sử một từ được nhận là 4063101012. Lặp lại những tính toán trên cho $S_1 = 9$, $S_2 = 7$, $S_3 = 10$, $S_4 = 2$ và $a = 0$, $b = 1$, $c = 5$. Bởi vì $a = 0$ nên từ Bước (5) của thuật toán suy ra có ít nhất ba lỗi được phát hiện và yêu cầu truyền tin lại.

KẾT LUẬN

Dựa trên cơ sở của lý thuyết đồng dư và trường hữu hạn, luận văn có mục đích tìm hiểu và trình bày những kiến thức, những kết quả cơ bản, sơ đẳng nhất của lý thuyết mã sửa sai. Ứng dụng lý thuyết số nói riêng, công cụ toán học nói chung (đại số tuyến tính, hệ đếm,…) cho phép nhận được nhiều kết quả quan trọng trong mã sửa sai, cũng như trong lý thuyết mật mã. Điều này có thể xem thêm trong các tài liệu tham khảo [1], [6], [7],...

Qua đây ta cũng thấy rằng, nhiều thành tựu toán học tưởng chừng như chỉ có ý nghĩa về lý thuyết, lại mang đến những ứng dụng quan trọng và bất ngờ trong thực tế. Nhiều kiến thức toán học tưởng chừng như sơ cấp (được giảng dạy trong trường phổ thông, thậm chí cấp Trung học cơ sở), nhưng có thể giảng dạy trong mối liên kết với những thành tựu ứng dụng mới trong tin học và đời sống. Hơn nữa, học sinh phổ thông hoàn toàn có đủ khả năng tiếp thu những kiến thức mới có nhiều ứng dụng này (hệ đếm, mã sửa sai, lý thuyết mật mã, lý thuyết đồ thị, toán rời rạc,…). Hy vọng rằng chương trình toán sơ cấp sẽ được bổ xung những mảng kiến thức toán này.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Phạm Huy Điển, Hà Huy Khoái, *Mã hóa thông tin: Cơ sở toán học và ứng dụng*, Nhà xuất bản Đại học Quốc gia, Hà Nội, 2004.

[2] Nguyễn Hữu Hoan, *Lý thuyết số*, Nhà xuất bản Đại học Sư phạm, Hà Nội, 2007.

[3] Bùi Doãn Khanh, Nguyễn Đình Thúc, *Mã hóa thông tin, Lý thuyết & ứng dụng*, Nhà xuất bản Lao động Xã hội, Thành phố Hồ Chí Minh, 2004.

[4] Hà Huy Khoái, *Nhập môn Số học thuật toán*, Nhà xuất bản Khoa học, Hà Nội, 1996.

[5] Hà Huy Khoái, Phạm Huy Điển, *Số học thuật toán: Cơ sở lý thuyết và Tính toán thực hành*, Nhà xuất bản Đại học Quốc Gia, Hà Nội, 2003.

Tiếng Anh

[6] Stephen Barnett, *Discrete Mathematics: Numbers and Beyond*, Addison Wesley Longman, Singapore, 1998.

[7] Sebastià Xambó-Descamps, *Block Error-Correcting Codes*, A Computational Primer, Springer-Verlag, 2000.

[8] Một số trang WEB và tạp chí Toán.