QRコードの符号化・復号アルゴリズム解説

— 学習読み物としての試み —

An introduction to the encoding-decoding algorithm of QR code

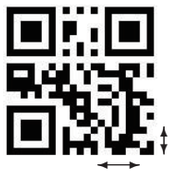
— An attempt to provide an educational reading —

ネットワーク情報学部　佐藤　創
School of Network and Information　Hajime SATO

Keywords : QR code, Reed-Solomon code, coding theory, linear algebra, sweep-out method

QR コード（Quick Response code）は部品管理のための 2 次元バーコードとして開発され, カメラ付き携帯電話と結合したことによって急速に普及しました。QR コードには短縮リード・ソロモン符号による誤り訂正機能が活用されています。自らプログラムを書けるほどにその符号化・復号アルゴリズムを理解するためには, 符号理論の本を読むのが常道ですが, それはなかなか難しいことです。ここではベクトルや行列などを学習した学生が読解できるような自己完結的な解説を試み,「なぜ未知の誤りの訂正ができるのか」という基本的な問いに丁寧に答えることにしました。このテーマは「線形代数」の格好の教材であると思っています。

春は名のみの風の寒さ
(訂正レベル L, $t_0 = 3$)



訂正可能 (長方形状の 24 セルがバーストノイズで反転)

復号誤り → 春は名ての風の映さ
(4 セルがランダムノイズで反転)

1　記号系について

1.1　情報の表現

情報を表現するには言語が基本となります。それぞれの言語には固有の記号系があり，その記号を 1 次元的に並べて情報を表現します。QR コードは見かけは 2 次元ですが，記号列を切り刻んで平面上に再配置したものにすぎません (付録 4 参照)。コンピュータやインターネットではよく知られている通り，情報は単純な 0 と 1 の列（ビット列）で表現されます。私たちの使う漢字かな混じり文とビット列の間には本質的な違いはなく，中間的な記号系の使用も含めて，相互変換が容易にできれば何も問題はありません。

記号列で表現された情報は伝達される過程で損傷を受け，受け取った記号列が変形することがあります。例えば QR コードは汚れたり一部が隠れたりします。元の情報を再現するにはどうすればよいでしょうか。日常会話では聞き取れないときすぐに聞き返えしますが，QR コードでは受け取った記号列をそれ自身で修復する方法がとられます。修復できるためには受け取る情報が冗長でなければなりません。つまり，情報を表現する記号列をもっと長いものに変換する必要があります。単なる反復もその素朴な例です。冗長さを加える変換を**符号化**と呼び，変形を受けた記号列から元の記号列を再現する逆の変換を**復号**と呼びます。また，符号を構成する記号の集合を**符号アルファベット**と呼ぶことにします。

1960 年頃イギリスの I. S. Reed と G. Solomon は代数学を応用して符号化と復号を巧妙に実現する方法を考案しました。それを**リード・ソロモン符号**（Reed-Solomon code，RS 符号）と呼びます。この符号では符号アルファベットとして加減乗除の演算が定義された有限の記号系を用いています。この記号系は“有限体”と呼ばれていますが, 以下ではその直感的な説明にとどめます。

1.2　加減乗除のできる有限の記号系

加減乗除の演算ができる有限の記号系は，記号の総数 q が素数 p の累乗 ($q = p^m$) でなければならいことが知られています。最も簡単なものは，0 と 1 の 2 個からなるもので，$1 + 1 = 0$（排他的論理和，XOR）が特徴です。

$q = 4 = 2^2$ の場合は関係 $\alpha^2 + \alpha + 1 = 0$ を満たす α を使った 0, 1, α, α^2 の 4 個からなり，次の演算表に従うものです。加算表の対角線上が 0 ですから減算は加算と同じで，除算は指数法則 $\alpha^i \times \alpha^j = \alpha^{i+j}$ と $\alpha^3 = 1$ から導かれます（$\alpha^3 - 1 = (\alpha - 1)(\alpha^2 + \alpha + 1) = 0$ に注意）。

+	0	1	α	α^2
0	0	1	α	α^2
1	1	0	α^2	α
α	α	α^2	0	1
α^2	α^2	α	1	0

×	0	1	α	α^2
0	0	0	0	0
1	0	1	α	α^2
α	0	α	α^2	1
α^2	0	α^2	1	α

QR コードでは符号アルファベットとして $q = 256 = 2^8$ 個の記号からなる記号系 $\mathcal{A}$ が使われます。個々の記号は

$$\alpha^8 + \alpha^4 + \alpha^3 + \alpha^2 + 1 = 0$$

という関係を満たす記号 α を用いて

$$\mathcal{A} = \{\, 0,\ 1,\ \alpha,\ \alpha^2,\ \cdots,\ \alpha^{254} \,\} \tag{1}$$

のように表し，この中に加減乗除を定義します。これらの記号を肩の数字 (指数) で区別するのは，乗除算に便利だからです。

QR コードは白と黒の小さな正方形 (セル) の並びで表現されますが，$\mathcal{A}$ の 1 個の記号は隣接する 8 セルに対応しています。

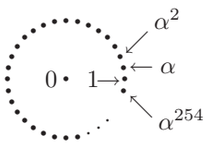


図 1　記号系 $\mathcal{A}$ のイメージ

記号 0 と 1 の役割は通常通りです。0 以外の記号の乗算を $\alpha^i \times \alpha^j = \alpha^{i+j}$ で, 除算を $\alpha^i / \alpha^j = \alpha^{i-j}$ で定義します。指数は周期 255 で循環する整数で表します（図 1）。例えば，$123 + 208 = 331$ は 76 に等しいので $\alpha^{123} \times \alpha^{208} = \alpha^{76}$，また，$123 - 208 = -85$ は 170 に等しいので $\alpha^{123} / \alpha^{208} = \alpha^{170}$ です。α^i の逆数は $\alpha^{-i} = \alpha^{255-i}$ です（$\alpha^{256} = \alpha^0 = 1$）。0 以外の記号 α^j を $j\,(0 \le j \le 254)$ で表すことを**指数表現**といいます（0 は指数表現できない）。

関係 (1) によりすべての記号は α の 7 次多項式（係数は 0 か 1）で表され，その係数列（長さ 8）を**ビット列表現**といいます。0 は 00000000，1 は 00000001，α は 00000010 などとなります。$\mathcal{A}$ の中での加算はビット列表現のビット毎 XOR から導かれます（付録 1 参照）。この記号系 $\mathcal{A}$ でも減算は加算と等しいという特性がありますが，演算の 3 法則（交換法則，結合法則，分配法則）が成り立ち，記号数の有限・無限の違いを除けば，実数と同様に扱うことができます。

QR 符号では $\mathcal{A}$ の記号をビット表現してそれを 8 個のセルの白黒に対応させますが, 以下の説明は指数表現で行います。

2　符号化

2.1　生成多項式と生成行列

私たちの伝えたい情報は符号アルファベット $\mathcal{A}$ の記号列 $a_1 a_2 \cdots\ (a_i \in \mathcal{A})$ で表現されます。$\mathcal{A}$ の記号列を長さ k のブロックに分割した各記号列 $\boldsymbol{u} = u_1 u_2 \cdots u_k$ を**情報語**と呼び，それを**符号語**と呼ばれる長さ n の記号列 $\boldsymbol{v} = v_1 v_2 \cdots v_n$ に変換する操作 $\boldsymbol{u} \to \boldsymbol{v}$ を**符号化** (encoding)，符号語の全体を (n, k) **符号**と呼びます。整数 n, k に関する制約は不等式 $0 < k < n \le q - 1$ だけです。ここでは符号化方式として特に，$\boldsymbol{v}$ の前方を $\boldsymbol{u}$ と一致させ，その後方に長さ $n - k$ の冗長部を付加するものを採用します。その逆変換 $\boldsymbol{v} \to \boldsymbol{u}$ は後方の $n - k$ 個の記号を単に除去するだけです。この方式は**組織符号化**（systematic coding）と呼ばれます。

符号を特定するには n, k 以外に，**生成多項式**（generator polynomial）と呼ばれる $n - k$ 次多項式 $g(x)$ を選ぶ必要があります。QR コードでは

$$g(x) = (x-1)(x-\alpha)\cdots(x-\alpha^{n-k-1}) = \prod_{i=0}^{n-k-1} (x - \alpha^i) \tag{2}$$

であり，これらによって定まる符号を『$1, \alpha, \cdots, \alpha^{n-k-1}$ **を根とする** (n, k) **短縮 RS 符号**』と呼びます。

式 (2) の $g(x)$ を展開した多項式を

$$g(x) = g_1\, x^{n-k} + g_2\, x^{n-k-1} + \cdots + g_{n-k+1}, \quad g_1 = 1 \tag{3}$$

とします（付録 2 参照）。その係数列 $\boldsymbol{g} = 1\, g_2 \cdots g_{n-k+1}$ から定まる $k \times n$ 行列

$$G_0 = \begin{bmatrix} 1 & g_2 & \cdots & g_{n-k+1} & 0 & \cdots & \cdots & 0 \\ 0 & 1 & g_2 & \cdots & g_{n-k+1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & g_2 & \cdots & g_{n-k+1} & 0 \\ 0 & \cdots & \cdots & 0 & 1 & g_2 & \cdots & g_{n-k+1} \end{bmatrix} \tag{4}$$

を**生成行列**（generator matrix）と呼びます。組織符号を得るには，行列 G_0 を “掃き出し法” によって $G = [\, I_k,\ P \,]$（I_k は $k \times k$ 単位行列）という標準形 (echlon form) に変形します。この変形を行列 T により $G = T\, G_0$ で表すことにすると，T は G_0 の第 k 列までの部分行列の逆行列にあたります。

$$G_0 \ \longrightarrow\ G = T\, G_0 = [\, I_k,\ P \,],$$

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n-k} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n-k} \\ \vdots & \vdots & & \vdots \\ p_{k,1} & p_{k,2} & \cdots & p_{k,n-k} \end{bmatrix}. \tag{5}$$

以下では，長さ n の記号列 $a_1 a_2 \cdots a_{n-1} a_n$ と $n - 1$ 次の多項式 $a_1 x^{n-1} + \cdots + a_{n-1} x + a_n$ とを対応させて，

記号列（ベクトル表現と呼ぶ）に関する記述を多項式で表現することがあります。例えば，式 (3) の生成多項式 $g(x)$ は係数列 $\boldsymbol{g}$ と対応します。

2.2 符号化アルゴリズム（入力 $\boldsymbol{u}$，出力 $\boldsymbol{v}$）

符号化 $\boldsymbol{u} \to \boldsymbol{v}$ は簡単で，次の 2 通り（掛け算の形と割り算の形）に表現できます。

[**ベクトル表現**]　情報語 $\boldsymbol{u}$ に対する符号語 $\boldsymbol{v}$ は標準形の生成行列 $G = [\,I_k,\ P\,]$ により，

$$\boldsymbol{v} = \boldsymbol{u}\,G = \boldsymbol{u}\cdot(\boldsymbol{u}\,P) \quad (\cdot \text{は記号列の連結を表す}). \tag{6}$$

[**多項式表現**]　情報多項式を $u(x) = \sum_{i=1}^{k} u_i\,x^{k-i}$，符号多項式を $v(x) = \sum_{i=1}^{n} v_i\,x^{n-i}$，$u(x)\,x^{n-k}$ を $g(x)$ で割った剰余多項式を $r(x)$ とすると，

$$v(x) = u(x)\,x^{n-k} - r(x). \tag{7}$$

[解説]　符号多項式 $v(x)$ は $g(x)$ で割り切れるので，

$$v(\alpha^{j-1}) = 0 \qquad (1 \le j \le n-k)$$

という性質があります。$v(x) = \tilde{u}(x)\,g(x)$ とし，商 $\tilde{u}(x)$ のベクトル表現を $\tilde{\boldsymbol{u}}$ とすると，$\boldsymbol{v} = \tilde{\boldsymbol{u}}\,G_0$ であり，剰余 $r(x)$ のベクトル表現 $\boldsymbol{r}$ により，$\boldsymbol{v} = \boldsymbol{u}\cdot\boldsymbol{r}$ とも表せます。$G = T\,G_0$ でしたから，$\boldsymbol{u} = \tilde{\boldsymbol{u}}\,T^{-1}$, $\boldsymbol{r} = \boldsymbol{u}\,P$ となります。

変形を受けても修復できるのは，冗長部 $\boldsymbol{r}$ にそのための情報が組み込まれるからです。

3 復号

3.1 誤りの発生

伝えるべき情報語 $\boldsymbol{u}$ は符号化され，符号語 $\boldsymbol{v}$ として記録・発信されますが，受信者 (例えば読み取り装置) に届く過程で変形する可能性があります。これを $\boldsymbol{v}$ に**誤り語** $\boldsymbol{e}$ (長さ n の記号列) が加えられて**受信語** $\boldsymbol{w}$ になったと考えることにします。すなわち，

$$\boldsymbol{w} = \boldsymbol{v} + \boldsymbol{e}, \qquad w_i = v_i + e_i \quad (i = 1, 2, \cdots, n). \tag{8}$$

つまり，誤り $w_i \neq v_i$ は $e_i \neq 0$ を意味します。$e_i \neq 0$ である i の個数 t を $\boldsymbol{w}$ の**誤り個数**と呼び，その i を**誤り位置** (error location)，その記号 $e_i\,(\neq 0)$ を**誤り記号** (error symbol) と呼びます。誤り語 $\boldsymbol{e}$ が分かれば，受信語 $\boldsymbol{w}$ の誤りを $\boldsymbol{w} + \boldsymbol{e} = \boldsymbol{v}$ と訂正することができます ($\boldsymbol{e} + \boldsymbol{e} = \boldsymbol{0}$ ですから)。

当然，誤り個数 t がある限度を超えると正しく復号できません。結論をいうと, その限界 t_0 は

$$t_0 = \left\lfloor \frac{n-k}{2} \right\rfloor \tag{9}$$

です ($\lfloor x \rfloor$ は, $x \ge 0$ のときは小数点以下切り捨てを表す) 。ここことは後で証明します (5.4 節の性質 7) が，取りあえずこれを前提に話を進めます。

例えば，冒頭の QR コードの図は $n = 26$, $k = 19$ の (n, k) 符号で，$t_0 = \lfloor 7/2 \rfloor = 3$ です。白黒反転が 24 セルでも 3 記号内に収まるもの (訂正可能，図中) と反転 4 セルが 4 記号に及ぶもの (訂正不能，図右) を示しました。

受信語 $\boldsymbol{w}$，誤り語 $\boldsymbol{e}$ に対応して $w(x) = \sum_{i=1}^{n} w_i\,x^{n-i}$ を**受信多項式**，$e(x) = \sum_{i=1}^{n} e_i\,x^{n-i}$ を**誤り多項式**と呼びます。定義より，$w(x) = v(x) + e(x)$ です。

受信語 $\boldsymbol{w}$ に t 個の誤りがあると仮定し，その位置が $i_1, i_2, \cdots, i_t$ であると仮定して定まる多項式

$$\begin{aligned} &\sigma(x) = (x - x_1)(x - x_2)\cdots(x - x_t), \\ &\text{ここに, } x_r = \alpha^{n-i_r} \quad (r = 1, 2, \cdots, t) \end{aligned} \tag{10}$$

を**誤り位置多項式**と呼び（x_r の指数部の表現に注意），それを展開したものを

$$\sigma(x) = \sigma_0\,x^t + \sigma_1\,x^{t-1} + \cdots + \sigma_{t-1}\,x + \sigma_t, \quad \sigma_0 = 1 \tag{11}$$

とおきます。この多項式が分かれば各誤り位置が分かり，それから各誤り記号も分かるという筋書きです 。

3.2 検査行列とシンドローム

受信語 $\boldsymbol{w}$ だけから情報語 $\boldsymbol{u}$ を推量する操作 $\boldsymbol{w} \to \boldsymbol{u}'$ が復号（decoding）です。正しく復号される ($\boldsymbol{u}' = \boldsymbol{u}$ となる) ためには，次の課題を解決する必要があります。

> 1° 誤り個数 t を求める。
> 2° 誤り位置 $i_1, i_2, \cdots, i_t$ を求める
> 　($1 \le i_1 < i_2 < \cdots < i_t \le n$ とする)。
> 3° 誤り記号 $e_{i_1}, e_{i_2}, \cdots, e_{i_t}$ を求める。

誤り語 $\boldsymbol{e}$ が分かれば受信語 $\boldsymbol{w}$ を $\boldsymbol{v}' = \boldsymbol{w} + \boldsymbol{e}$ に訂正し，分からない場合は $\boldsymbol{v}' = \boldsymbol{w}$ として，復号手順を終ります：

4° $\boldsymbol{v}'$ の長さ k の前部を情報語 $\boldsymbol{u}'$ とする。

さて, 第 $n-1$ 列に生成多項式の根が並ぶ次の $(n-k)\times n$ 行列 H を，**検査行列** (check matrix) と呼びます[1]：

$$H = \begin{bmatrix} 1 & \cdots & 1 & 1 \\ \alpha^{n-1} & \cdots & \alpha & 1 \\ \vdots & & \vdots & \vdots \\ \alpha^{(n-k-1)(n-1)} & \cdots & \alpha^{n-k-1} & 1 \end{bmatrix}. \tag{12}$$

[1] 検査行列 H の標準形は，式 (5) の行列 P を用いて $H_1 = \left[P^{\mathrm{T}},\ I_{n-k}\right]$ と表され，$G\,H_1^{\mathrm{T}} = P + P = O$（ゼロ行列）となる.

課題1°, 2°, 3° の解決には，以下に定義される長さ $n-k$ の記号列 $\boldsymbol{s} = s_1\, s_2 \cdots s_{n-k}$ が重要な鍵となります。この記号列 $\boldsymbol{s}$ を**シンドローム** (syndrome，徴候，症候群の意味) と名付けます。

[ベクトル表現]　検査行列 H により，

$$\boldsymbol{s} = \boldsymbol{w}\, H^{\mathrm{T}} \qquad (H^{\mathrm{T}} \text{は } H \text{ の転置行列}). \tag{13}$$

[多項式表現]　受信多項式 $w(x) = \sum_{i=1}^{n} w_i\, x^{n-i}$ より，

$$s_j = w(\alpha^{j-1}) \qquad (j = 1, 2, \cdots, n-k). \tag{14}$$

[解説]　どちらにしても，$s_j = \sum_{i=1}^{n} w_i\, \alpha^{(j-1)(n-i)}$ $(1 \le j \le n-k)$ の計算をします。

シンドロームは受信語の検査結果を表します。もし誤りがなければ，$e(x) = 0 \Leftrightarrow w(x) = v(x)$ ですから $s_j = v(\alpha^{j-1}) = 0\,(1 \le j \le n-k)$ です。逆は必ずしも真ではありませんが，シンドローム $\boldsymbol{s}$ は

$$\begin{cases} \boldsymbol{s} \neq \boldsymbol{0} & \text{ならば} \quad \boldsymbol{e} \neq \boldsymbol{0} \quad (\text{何らかの誤りがある}), \\ \boldsymbol{s} = \boldsymbol{0} & \text{ならば} \quad \boldsymbol{e} \approx \boldsymbol{0} \quad (\text{多分，誤りがない})^{2} \end{cases}$$

ということを知らせてくれます。$\boldsymbol{s} \neq \boldsymbol{0}$ のときは誤り語 $\boldsymbol{e}$ に関するさらに詳しい情報が得られます。

シンドローム $\boldsymbol{s}$ を次のように配置した $t_0 \times t_0$ 行列を**シンドローム行列**（syndrome matrix）と呼びます。

$$S = \begin{bmatrix} s_1 & s_2 & \cdots & s_{t_0} \\ s_2 & s_3 & \cdots & s_{t_0+1} \\ \vdots & \vdots & & \vdots \\ s_{t_0} & s_{t_0+1} & \cdots & s_{2t_0-1} \end{bmatrix}. \tag{15}$$

3.3　復号アルゴリズム（入力 $\boldsymbol{w}$，出力 $\boldsymbol{u}'$）

『$1, \alpha, \cdots, \alpha^{n-k-1}$ を根とする (n, k) 短縮 RS 符号』の復号アルゴリズムを記します。これは受信語 $\boldsymbol{w}$ に対する課題 1°, 2°, 3° の解決手順を述べることにほかなりません。

(1°) 式 (13) によりシンドローム $\boldsymbol{s}$ を計算する。$\boldsymbol{s} = \boldsymbol{0}$ ならば**誤りなし**と判定し，$\boldsymbol{v}' = \boldsymbol{w}$ として (4°) へ。$\boldsymbol{s} \neq \boldsymbol{0}$ ならば，誤り個数 t は式 (15) のシンドローム行列の階数[3] rank S に等しい。

実際の手順は次の (2°) と一体化する。

(2°) シンドローム行列 S に下のような 1 列を加えた拡大行列 $\widetilde{S}$ に対して掃き出し計算を行う。第 t 行まで対角線上に 1 が並び，それ以下の行がすべて 0 になったとき，その t が誤り個数である。ただし，対角線上の 1 が第 t_0 行まで並んだ場合は，t_0 が誤り個数とは限らない（決定は (2°) の最後）。掃き出し計算の結果，式 (11) の誤り位置多項式 $\sigma(x)$ の係数 $\sigma_t, \sigma_{t-1}, \cdots, \sigma_1$ が第 $t+1$ 列に並ぶ。

$$\widetilde{S} = \begin{bmatrix} s_1 & s_2 & \cdots & s_{t_0} & s_{t_0+1} \\ s_2 & s_3 & \cdots & s_{t_0+1} & s_{t_0+2} \\ \vdots & \vdots & & \vdots & \vdots \\ s_{t_0} & s_{t_0+1} & \cdots & s_{2t_0-1} & s_{2\,t_0} \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 0 & \cdots & 0 & \sigma_t & * & * \\ 0 & 1 & \cdots & 0 & \sigma_{t-1} & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & \sigma_1 & * & * \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

誤り位置多項式 $\sigma(x)$ の変数 x に記号 α^j $(j = n-1, \cdots, 1, 0)$ を次々に代入して，

$$\sigma(\alpha^j) = 0 \tag{17}$$

を満たす α^j を見つけて順に誤り位置 $i_r = n - j$ $(r = 1, 2, \cdots, t)$ とする。ただし，$t = t_0$ のとき式 (17) を満たす記号 α^j が見つからない場合は，誤り個数は t_0 より多く，**訂正不能**と判定し，$\boldsymbol{v}' = \boldsymbol{w}$ として (4°) へ。

(3°) 誤り位置について $k < i_1$ ならば $\boldsymbol{w}$ の情報語部分は訂正の必要がないので，$\boldsymbol{v}' = \boldsymbol{w}$ として (4°) へ。そうでなければ，誤り記号 $e_{i_1}, e_{i_2}, \cdots, e_{i_t}$ を連立 1 次方程式

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_t \\ \vdots & \vdots & & \vdots \\ x_1^{t-1} & x_2^{t-1} & \cdots & x_t^{t-1} \end{bmatrix} \begin{bmatrix} e_{i_1} \\ e_{i_2} \\ \vdots \\ e_{i_t} \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_t \end{bmatrix} \tag{18}$$

(式 (10) より $x_r = \alpha^{n-i_r}$)

の解として求め，$\boldsymbol{w}$ の誤りを $v'_{i_r} = w_{i_r} + e_{i_r}$ $(r = 1, 2, \cdots, t)$ と訂正したものを符号語 $\boldsymbol{v}'$ とする。

(4°) 記号列 $\boldsymbol{v}'$ の長さ k の前部を情報語 $\boldsymbol{u}'$ とする（訂正不能の場合も $\boldsymbol{u}'$ は「参考情報」になる）。

4　計算例

『$1, \alpha, \alpha^2, \alpha^3$ を根とする $(n = 10, k = 6)$ 短縮 RS 符号』を例として，変形した符号語がどのように修復されるかを詳細に見てみます。アルゴリズムの遂行にどんな計算が必要か，具体的に知ることができます。

符号化に先立って，生成多項式 $g(x)$，生成行列 G を求めます。

[2]「誤りがない」といっても他の符号語に一致するほど多くの誤りが生じた場合と区別できない（後述）.

[3] 一般に，行列の線形独立な行ベクトルの個数は線形独立な列ベクトルの個数と一致し，これを行列の階数 (rank) と呼ぶ.

生成多項式の次数は $n-k=4$ ですから $g(x)=(x-1)(x-\alpha)(x-\alpha^2)(x-\alpha^3)$ で，それを展開した係数が必要です。この解説では記号の加算演算のプロセス（付録 2 参照）を省略し，その結果を記します：

$$g(x)=x^4+\alpha^{75}x^3+\alpha^{249}x^2+\alpha^{78}x+\alpha^6,$$
$$\boldsymbol{g}=1\,\alpha^{75}\,\alpha^{249}\,\alpha^{78}\,\alpha^6.$$

係数列 $\boldsymbol{g}$ をシフトしながら並べた生成行列 G_0 から，掃き出し法によって得られる標準形 (5) の G の部分行列 P を示します：

$$G_0\ \to\ G=[\,I_6,P\,],\ P=\begin{bmatrix}\alpha^{134}&\alpha^{157}&\alpha^{71}&\alpha^{211}\\ \alpha^{205}&\alpha^{96}&\alpha^{177}&\alpha^{149}\\ \alpha^{143}&\alpha^{49}&\alpha^{253}&\alpha^{137}\\ \alpha^{131}&\alpha^{198}&\alpha^{162}&\alpha^{169}\\ \alpha^{163}&\alpha^{189}&\alpha^{59}&\alpha^{81}\\ \alpha^{75}&\alpha^{249}&\alpha^{78}&\alpha^{6}\end{bmatrix}.$$

[**符号化**]　試みに，情報語，情報多項式を次のように定めます (例として円周率から 3.1415926535 を利用)。

$$u(x)=\alpha^3x^5+\alpha^{14}x^4+\alpha^{15}x^3+\alpha^{92}x^2+\alpha^{65}\,x+\alpha^{35},$$
$$\boldsymbol{u}=\alpha^3\,\alpha^{14}\,\alpha^{15}\,\alpha^{92}\,\alpha^{65}\,\alpha^{35}.$$

情報多項式 $u(x)$ を x^4 倍した多項式 $u(x)\,x^4$ を生成多項式 $g(x)$ で割った余り $r(x)$ を計算した結果は

$$r(x)=\alpha^{240}x^3+\alpha^{168}x^2+\alpha^{152}x+\alpha^{204},$$
$$\boldsymbol{r}=\alpha^{240}\,\alpha^{168}\,\alpha^{152}\,\alpha^{204}$$

となります。したがって，符号多項式 (7) と符号語 $\boldsymbol{v}=\boldsymbol{u}\cdot\boldsymbol{r}$ は次の通りです。

$$v(x)=u(x)\,x^4-(\alpha^{240}x^3+\alpha^{168}x^2+\alpha^{152}x+\alpha^{204}),$$
$$\boldsymbol{v}=\alpha^3\,\alpha^{14}\,\alpha^{15}\,\alpha^{92}\,\alpha^{65}\,\alpha^{35}\,\alpha^{240}\,\alpha^{168}\,\alpha^{152}\,\alpha^{204}.$$

同じ結果は，生成行列 G により式 (6) の行列計算 $\boldsymbol{v}=\boldsymbol{u}\,G=\boldsymbol{u}\cdot\boldsymbol{u}\,P$ からも得られます。

[**誤りの発生**]　一例として，受信語

$$\boldsymbol{w}=\underline{\alpha^4}\,\alpha^{14}\,\alpha^{15}\,\alpha^{92}\,\alpha^{65}\,\alpha^{35}\,\alpha^{240}\,\underline{\alpha^{236}}\,\alpha^{152}\,\alpha^{204}$$

を受け取ったものと仮定しましょう。$\boldsymbol{v}$ を知らないので誤り個数，誤り位置，誤り記号はすぐには分かりません。実は 2 個の誤りがあって，$t_0=(n-k)/2=2$ ですから，この誤りは訂正可能のはずです。

復号に先立って検査行列 (12) を記します:

$$H=\begin{bmatrix}1&1&1&1&1&1&1&1&1&1\\ \alpha^9&\alpha^8&\alpha^7&\alpha^6&\alpha^5&\alpha^4&\alpha^3&\alpha^2&\alpha&1\\ \alpha^{18}&\alpha^{16}&\alpha^{14}&\alpha^{12}&\alpha^{10}&\alpha^8&\alpha^6&\alpha^2&\alpha^2&1\\ \alpha^{27}&\alpha^{24}&\alpha^{21}&\alpha^{18}&\alpha^{15}&\alpha^{12}&\alpha^9&\alpha^4&\alpha^3&1\end{bmatrix}.$$

[**復号**]　誤りを訂正するプロセスを示します。まずシンドロームを計算すると，式 (14) により

$$\begin{cases}s_1=\ w(1)&=\sum_{i=1}^{10}w_i&=\alpha^{69},\\ s_2=\ w(\alpha)&=\sum_{i=1}^{10}w_i\,\alpha^{10-i}&=\alpha^{180},\\ s_3=\ w(\alpha^2)&=\sum_{i=1}^{10}w_i\,\alpha^{2(10-i)}&=\alpha^{196},\\ s_4=\ w(\alpha^3)&=\sum_{i=1}^{10}w_i\,\alpha^{3(10-i)}&=\alpha^{89}\end{cases}$$

となります。同じ結果は検査行列 H により，式 (13) の $\boldsymbol{s}=\boldsymbol{w}\,H^{\mathrm{T}}$ を計算しても得られます。

以下，シンドローム $\boldsymbol{s}=\alpha^{69}\,\alpha^{180}\,\alpha^{196}\,\alpha^{89}$ からすべてが得られることを示します。

拡大したシンドローム行列 $\widetilde{S}$ の掃き出し法による変形は

$$\widetilde{S}=\begin{bmatrix}s_1&s_2&s_3\\ s_2&s_3&s_4\end{bmatrix}=\begin{bmatrix}\alpha^{69}&\alpha^{180}&\alpha^{196}\\ \alpha^{180}&\alpha^{196}&\alpha^{89}\end{bmatrix}$$
$$\longrightarrow\quad\begin{bmatrix}1&0&\alpha^{11}\\ 0&1&\alpha^{114}\end{bmatrix},\qquad\operatorname{rank}S=2$$

となるので，(1°) 誤り個数が $t=2$ (ただし，$t_0=2$ だから誤り個数は暫定) であることと，誤り位置多項式 $\sigma(x)$ の係数 $\sigma_1=\alpha^{114},\sigma_2=\alpha^{11}$ が決定し，

$$\sigma(x)=x^2+\alpha^{114}x+\alpha^{11}$$

を得ます。方程式 $\sigma(x)=0$ には解 $x=\alpha^9$, $x=\alpha^2$ があることから誤り個数 $t=2$ が確定し，(2°) 誤り位置

$$i_1=n-9=1,\qquad i_2=n-2=8$$

が判明します。(3°) 誤り記号は，シンドローム s_1, s_2 の定義による連立 1 次方程式 (18)

$$\begin{bmatrix}1&1\\ \alpha^9&\alpha^2\end{bmatrix}\begin{bmatrix}e_1\\ e_8\end{bmatrix}=\begin{bmatrix}\alpha^{69}\\ \alpha^{180}\end{bmatrix}$$

を解いて (ここでも掃き出し計算)，$e_1=\alpha^{28}$, $e_8=\alpha^{185}$ であることが分かります。

最終的に，(4°) 誤り訂正 $w_1\to\alpha^4+\alpha^{28}=\alpha^3$, $w_8\to\alpha^{236}+\alpha^{185}=\alpha^{168}$ により復元された符号語 $\boldsymbol{v}$ から元の情報語 $\boldsymbol{u}=\alpha^3\,\alpha^{14}\,\alpha^{15}\,\alpha^{92}\,\alpha^{65}\,\alpha^{35}$ が再現されます（この場合，w_1 の訂正だけで十分)。

• **誤り個数が 1 の場合**　このときは少ない手数で誤りが訂正できます。例えば，受信語を

$$\boldsymbol{w}=\underline{\alpha^4}\,\alpha^{14}\,\alpha^{15}\,\alpha^{92}\,\alpha^{65}\,\alpha^{35}\,\alpha^{240}\,\alpha^{168}\,\alpha^{152}\,\alpha^{204}$$

と仮定しましょう。シンドロームを計算すると

$$s_1=\alpha^{28},\quad s_2=\alpha^{37},\quad s_3=\alpha^{46},\quad s_4=\alpha^{55}$$

となります。これより拡大シンドローム行列は

$$\widetilde{S} = \begin{bmatrix} s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \end{bmatrix} = \begin{bmatrix} \alpha^{28} & \alpha^{37} & \alpha^{46} \\ \alpha^{37} & \alpha^{46} & \alpha^{55} \end{bmatrix}$$
$$\longrightarrow \quad \begin{bmatrix} 1 & \alpha^9 & \alpha^{18} \\ 0 & 0 & 0 \end{bmatrix}, \qquad \operatorname{rank} S = 1$$

と変形されるので，誤り個数 $t = 1$ と $\sigma_1 = \alpha^9$ が分かり，誤り多項式 $\sigma(x) = x + \alpha^9$ が決定します。直ちに $\sigma(x) = 0$ の解 $x_1 = \alpha^9$ とそれから誤り位置 $i_1 = 10 - 9 = 1$ が求まります。したがって，誤り記号は方程式 (18) より $e_1 = s_1 = \alpha^{28}$ ですから，w_1 の誤りが $w_1 + e_1 = \alpha^3 = v_1$ と訂正されます。

● **訂正不能の例**　誤り個数が 3 である場合，例えば受信語が

$$\boldsymbol{w} = \underline{\alpha^4}\,\alpha^{14}\,\alpha^{15}\,\alpha^{92}\,\alpha^{65}\,\alpha^{35}\,\alpha^{240}\,\underline{\alpha^{236}}\,\alpha^{152}\,\underline{\alpha^{36}}$$

のときはどうなるでしょうか。シンドローム $\boldsymbol{s}$ の計算後，拡大シンドローム行列 $\widetilde{S}$ の掃き出し計算

$$\widetilde{S} = \begin{bmatrix} \alpha^{170} & \alpha^{186} & \alpha^{29} \\ \alpha^{186} & \alpha^{29} & \alpha^{193} \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 0 & \alpha^{167} \\ 0 & 1 & \alpha^{245} \end{bmatrix}$$

であるとどうでしょうか。今度もシンドローム $\boldsymbol{s}$ から得られる行列 $\widetilde{S}$ の掃き出し計算

$$\widetilde{S} = \begin{bmatrix} \alpha^{154} & \alpha^{7} & \alpha^{62} \\ \alpha^{7} & \alpha^{62} & \alpha^{232} \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 0 & \alpha^{4} \\ 0 & 1 & \alpha^{51} \end{bmatrix}$$

の結果，暫定誤り個数 $t = 2$ と誤り位置多項式 $\sigma(x) = x^2 + \alpha^{51}x + \alpha^4$ を得ます。今度は方程式 $\sigma(x) = 0$ には解 $x = \alpha^3, x = \alpha$ があるので誤り個数 $t = 2$ が確定し，誤り位置 $i_1 = n - 3 = 7, \quad i_2 = n - 1 = 9$ が判明します。$k < i_1 < i_2$ ですから誤り記号 e_7, e_9 を求めるまでもなく，$\boldsymbol{w}$ の前部が直ちに情報語

$$\boldsymbol{u}' = \underline{\alpha^4}\,\alpha^{14}\,\alpha^{15}\,\alpha^{92}\,\alpha^{65}\,\alpha^{35}$$

として復号されます。これは元の情報語 $\boldsymbol{u}$ ではなく，訂正に失敗したことも識別できません。

この場合，受信語 $\boldsymbol{w}$ は，符号語 $\boldsymbol{v}$ を基準にすると 3 個の誤り（$e_1 = \alpha^{28}, e_8 = \alpha^{185}, e_{10} = \alpha^{239}$）が生じているので訂正できないはずです。しかし，別の符号語 $\boldsymbol{v}' = \alpha^4\,\alpha^{14}\,\alpha^{15}\,\alpha^{92}\,\alpha^{65}\,\alpha^{35}\,\alpha^{119}\,\alpha^{236}\,\alpha^{246}\,\alpha^{236}$ を基準にすると 2 個の誤り $e_7 = \alpha^{162}, e_9 = \alpha^{99}$ になるので，誤り訂正は $\boldsymbol{v}'$ の方へ向かったのでした。

5　なぜ誤りが訂正できるか

誤り個数が t_0 以下ならば，3 章で述べたアルゴリズムにより誤りが訂正できることを 4 章の計算例で示しました。以下では，なぜこのアルゴリズムによって誤りが訂正できるのか，その理由を明らかにします。

まず符号語とシンドロームの性質を述べ，次にアルゴリズムがどのように導かれるかを示します。

5.1　符号語・シンドロームの性質

$q = 2^8$ 個の記号からなる記号系 $\mathcal{A}$ の上の (n, k) 短縮 RS 符号において，情報語 $\boldsymbol{u}$ は長さ k の任意の記号列です。q^k 個の情報語全体を $\mathcal{A}^k$ で表すことにしましょう。記号 a と情報語 $\boldsymbol{u}$ の積 $a\,\boldsymbol{u}$ を $\boldsymbol{u}$ の各要素の a 倍からなる列で定義し，2 つの情報語 $\boldsymbol{u}_1$, $\boldsymbol{u}_2$ の和 $\boldsymbol{u}_1 + \boldsymbol{u}_2$ を各要素ごとの和からなる列と定義すると，$\mathcal{A}^k$ は k 次元ベクトル空間を形成します。

同様に，符号語は長さ n の記号列で，符号語の全体を C で表すと $a \in \mathcal{A}$, $\boldsymbol{v} \in C$ ならば $a\,\boldsymbol{v} \in C$，かつ，$\boldsymbol{v}_1, \boldsymbol{v}_2 \in C$ ならば $\boldsymbol{v}_1 + \boldsymbol{v}_2 \in C$ が成り立つので，符号 C は n 次元ベクトル空間 $\mathcal{A}^n$ の中で k 次元部分空間を形成します。このようにベクトル空間を形成する符号を一般に**線形符号** (linear code) と呼びます。

長さ n の記号列がすべて符号語というわけではなく，符号語の性質として次が重要です。

性質 1　長さ n の記号列 $\boldsymbol{w}$ が符号語であるための必要十分条件は，

$$\text{検査行列 } H \text{ に対して} \quad \boldsymbol{w}\,H^{\mathrm{T}} = \boldsymbol{0}. \tag{19}$$

その証明は，記号列 $\boldsymbol{w} \in \mathcal{A}^n$ に対応する多項式を $w(x)$ として次の同値関係をたどることで得られます。

$$\begin{aligned} &\boldsymbol{w}\text{は符号語} \quad \Leftrightarrow \quad w(x) \text{ は } g(x) \text{ で割り切れる} \\ \Leftrightarrow \quad & j = 1, 2, \cdots, n-k \text{ に対して } w(\alpha^{j-1}) = 0 \\ \Leftrightarrow \quad & j = 1, 2, \cdots, n-k \text{ に対して} \sum_{i=1}^{n} w_i\,\alpha^{(j-1)(n-i)} = 0 \\ \Leftrightarrow \quad & \boldsymbol{w}\,H^{\mathrm{T}} = 0. \end{aligned}$$

発信された符号語 $\boldsymbol{v}$，受信語 $\boldsymbol{w}$，誤り語 $\boldsymbol{e} = \boldsymbol{w} - \boldsymbol{v}$，および，$\boldsymbol{w}$ のシンドローム $\boldsymbol{s} = \boldsymbol{w}\,H^{\mathrm{T}}$ に関して，性質 1 ($\boldsymbol{v}\,H^{\mathrm{T}} = \boldsymbol{0}$) から次の性質が導かれます。

性質 2　符号語 $\boldsymbol{v}$ に対する受信語が $\boldsymbol{w} = \boldsymbol{v} + \boldsymbol{e}$ ならば，

$$\boldsymbol{w} \text{ のシンドローム } \boldsymbol{s} = \boldsymbol{w}\,H^{\mathrm{T}} \text{ について} \qquad \boldsymbol{s} = \boldsymbol{e}\,H^{\mathrm{T}}. \tag{20}$$

受信語 $\boldsymbol{w}$ のシンドローム $\boldsymbol{s}$ が誤り語 $\boldsymbol{e}$ だけから決まるところがポイントです。

次の図は，シンドロームの性質を表したもので，線形代数の教科書によく登場する概念図です。

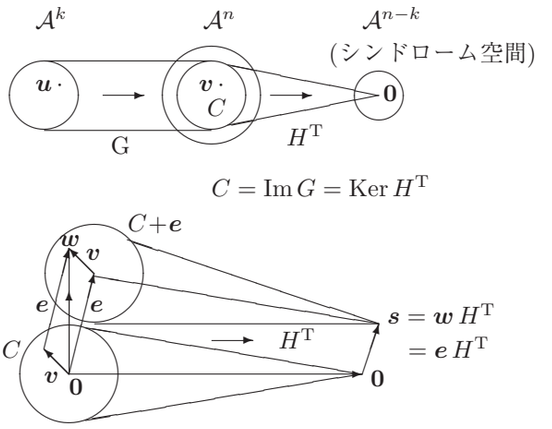


図 2　符号化，シンドローム計算を線形写像と考えた概念図

ところで，シンドローム行列 S の $r \times r$ 部分行列を

$$S_r = \begin{bmatrix} s_1 & s_2 & \cdots & s_r \\ s_2 & s_3 & \cdots & s_{r+1} \\ \vdots & \vdots & & \vdots \\ s_r & s_{r+1} & \cdots & s_{2r-1} \end{bmatrix} \qquad (1 \le r \le t_0) \tag{21}$$

で表すことにする（$S_{t_0} = S$）と，行列 S_r は次のように表現できます（証明は簡単[4]，性質 2 による）。

$$S_r = H_r\, E\, H_r^{\mathrm{T}}, \quad \text{ここに, } E = \begin{bmatrix} e_1 & 0 & \cdots & 0 \\ 0 & e_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e_n \end{bmatrix},$$

$$H_r = \begin{bmatrix} 1 & \cdots & 1 & 1 \\ \alpha^{n-1} & \cdots & \alpha & 1 \\ \vdots & & \vdots & \vdots \\ \alpha^{(r-1)\,(n-1)} & \cdots & \alpha^{r-1} & 1 \end{bmatrix}. \tag{22}$$

行列 H_r は検査行列 H の第 r 行までの $r \times n$ 小行列ですから，$\mathrm{rank}\, H_r \le r$ です。また，対角行列 E の階数は誤り個数に等しいので $\mathrm{rank}\, E = t$ です。したがって，

$$\mathrm{rank}\, S_r \le \min\{r,\, t\} \qquad (1 \le r \le t_0) \tag{23}$$

が成り立ちます[5]。さらに，式 (22) は $r = t\,(\le t_0)$ のとき，対角行列 E から $e_i = 0$ である行や列を除き，対応する H_t の列を取り除き，$x_j = \alpha^{n-i_j}$, $y_j = e_{i_j}$ $(j = 1, 2, \cdots, t)$ とおけば

$$S_t = X\, Y\, X^{\mathrm{T}}, \qquad \text{ここに, } Y = \begin{bmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & y_t \end{bmatrix},$$

$$X = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_t \\ \vdots & \vdots & & \vdots \\ x_1^{t-1} & x_2^{t-1} & \cdots & x_t^{t-1} \end{bmatrix} \tag{24}$$

となります。行列 X の行列式は次の形になることが知られています（ファンデルモンドの行列と呼ばれる）。

$$|X| = \prod_{i>j} (x_i - x_j) \quad (i > j \text{ であるすべての差 } x_i - x_j \text{ の積}).$$

x_j はすべて異なるので $|X| \neq 0$ であり，式 (24) 右辺の 3 つの行列は正則ですから[6]，$|S_t| \neq 0$，すなわち，$\mathrm{rank}\, S_t = t$ であることが分かります。したがって，不等式 (23) と合わせると次の性質が導かれます。

性質 3　受信語 $\boldsymbol{w}$ の誤り個数 t に関して，

$$0 \le t \le t_0 \text{ ならば} \qquad t = \mathrm{rank}\, S. \tag{25}$$

ただし，$t > t_0$ のときも $\mathrm{rank}\, S = t_0$ ですから，このときには注意が必要です。

実際にシンドローム行列 S の階数 $\mathrm{rank}\, S$ を計算するには，3.3 節で述べたように，拡大した行列 $\widetilde{S}$ の掃き出し (16) を行うのがよいでしょう。誤り位置多項式の係数 $\sigma_1, \sigma_2, \cdots, \sigma_t$ が同時に求まりますから。

以上，3 つの性質に基づいてアルゴリズムの正しさが以下で示されます。

5.2　素朴な考察

最初に，素朴な疑問に答えておきます。その疑問は，「誤りの基準となる符号語 $\boldsymbol{v}$ を知らないのだから，受信語 $\boldsymbol{w}$ の誤りなど，わかるはずがない。そもそも誤りの訂正は不可能ではないか」というもので，根本的な問題提起でもあります。

もちろん何も分からないわけではなく，性質 2 よりシンドローム $\boldsymbol{s}$ が $\mathbf{0}$ でなければ受信語 $\boldsymbol{w}$ に誤りが含まれることは確実です。しかし，別の符号語 $\boldsymbol{v}_1$ に対して

[4] S_r の (i,j) 成分は，$s_{i+j-1} = w(\alpha^{i+j-2}) = e(\alpha^{i+j-2}) = \sum_{r=1}^{n} e_r\, \alpha^{(i+j-2)\,(n-r)} = \sum_{r=1}^{n} \alpha^{(i-1)\,(n-r)}\, e_r\, \alpha^{(j-1)\,(n-r)}$
$= (\alpha^{(i-1)\,(n-1)}, \cdots, \alpha^{i-1}, 1) \left[\delta_{i,j}\, e_i\right] (\alpha^{(j-1)\,(n-1)}, \cdots, \alpha^{j-1}, 1)^{\mathrm{T}}$.
ここに，$\delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$ で，$\left[\delta_{i,j}\, e_i\right]$ は対角行列 E.

[5] 一般に，行列の積の階数は各行列の階数の最小値以下，すなわち，$\mathrm{rank}\, A\, B \le \min\{\mathrm{rank}\, A, \mathrm{rank}\, B\}$ である．

[6] 一般に，正方行列の積の行列式は各行列の行列式の積に等しい．すなわち，$|A\, B| = |A|\, |B|$ である．

$\boldsymbol{w}=\boldsymbol{v}+\boldsymbol{e}=\boldsymbol{v}_1+\boldsymbol{e}_1\,(\boldsymbol{e}\neq\boldsymbol{e}_1)$ ならば $\boldsymbol{s}=\boldsymbol{e}\,H^{\mathrm{T}}=\boldsymbol{e}_1\,H^{\mathrm{T}}$ ですから，シンドローム $\boldsymbol{s}$ からは 2 つの誤り語 $\boldsymbol{e}$, $\boldsymbol{e}_1$ の識別はできません。とくに $\boldsymbol{e}_1=\boldsymbol{0}\,(\boldsymbol{e}=\boldsymbol{v}_1-\boldsymbol{v})$ のとき，つまり発信された符号語 $\boldsymbol{v}$ が $\boldsymbol{v}_1$ になるまで変化した場合，$\boldsymbol{s}=\boldsymbol{0}$ ですからこの “変身” $\boldsymbol{v}\to\boldsymbol{v}_1$ は全く検出できないことになります。したがって,「素朴な疑問」の指摘は尤もで，受信語 $\boldsymbol{w}$ を訂正して発信された符号語 $\boldsymbol{v}$ を常に正しく特定することは原理的に不可能です。

この解説ではそれを踏まえた上で次の方針に従います。

> 「誤り」は，未知の符号語を基準に考えるのではなく，受信語 $\boldsymbol{w}$ を基準に考える。すなわち，$\boldsymbol{w}$ からの距離 (異なり記号数) が t_0 以下のところに符号語 $\boldsymbol{v}$ があれば (あるとすればただ 1 つ)，それを発信された符号語とみなす。発信された符号語との一致は確かめようがないので，当面の問題から除外する。$\boldsymbol{w}$ からの距離が t_0 以下のところに符号語がない場合は，訂正を断念する。

実際の応用に際しては，高い確率で誤り個数が t_0 以下であると想定できる場合が多く，このときは発信された符号語が高い確率で再現できます。

したがって問題の核心は，与えられた受信語 $\boldsymbol{w}$ に対して，$\boldsymbol{w}$ から t_0 以下の距離にある符号語を探し出す (なければないと断定する) アルゴリズムの実現とその正しさの保証である，ということになります。

試みに，このアルゴリズムを素朴に発想してみます。$\boldsymbol{s}\neq\boldsymbol{0}$ とします。もし 1 つの記号が誤ったものとすると，その位置 i の可能性は $1\sim n$ の n 通り，かつ，正しい値は w_i 以外の $\mathcal{A}$ の記号ですから可能性は $q-1=255$ 通りで，総数 $255\,n$ 個の記号列 $\boldsymbol{e}=0\cdots0\,e_i\,0\cdots0$ の中からシンドローム $\boldsymbol{s}=(\boldsymbol{w}+\boldsymbol{e})\,H^{\mathrm{T}}$ が $\boldsymbol{0}$ になるものをコンピュータを使って探せばよいことになります。見つからなければ次は 2 つの記号に誤りがあると仮定し，${}_nC_2\times(q-1)^2$ 個の候補の中での探索が続きます。さらに探索の範囲を広げると作業量が膨大になり，やがて限界に達します。

この意味で 3 章に記したアルゴリズムは素晴らしいものです。正しく，かつ，迅速にこの作業を実現していることを以下で確かめましょう。

5.3　アルゴリズムの導出 (課題の解決)

改めて出発点から考えてみましょう。$\boldsymbol{s}=\boldsymbol{0}$ のときは「誤りなし」で決着ですから，以下 $\boldsymbol{s}\neq\boldsymbol{0}$ とします。訂正に関して与えられるすべての情報は，受信語 $\boldsymbol{w}$ とそれから導かれるシンドローム $\boldsymbol{s}=s_1s_2\cdots s_{n-k}$ です。誤り語 $\boldsymbol{e}$ が分かれば $\boldsymbol{w}$ は $\boldsymbol{w}+\boldsymbol{e}$ に訂正できます。$\boldsymbol{s}$ から $\boldsymbol{e}$ を導くことが求められています。

さて，$\boldsymbol{s}$ の値は性質 2，すなわち式 (20) により，未知の誤り語 $\boldsymbol{e}$ によって

$$\begin{cases} e_1 & +e_2 & +\cdots+e_n=s_1,\\ e_1\,\alpha^{n-1} & +e_2\,\alpha^{n-2} & +\cdots+e_n=s_2,\\ e_1\,\alpha^{2(n-1)} & +e_2\,\alpha^{2(n-2)} & +\cdots+e_n=s_3,\\ \quad\vdots & \quad\vdots & \qquad\vdots\quad\vdots\\ e_1\,\alpha^{(n-k-1)(n-1)} & +e_2\,\alpha^{(n-k-1)(n-2)} & +\cdots+e_n=s_{n-k} \end{cases} \tag{26}$$

と表現できます。式 (26) を未知数 $e_1,e_2,\cdots,e_n$ に関する連立 1 次方程式と考えると，等式の数が少なくて解くことができません。ところが，$e_1,e_2,\cdots,e_n$ の中に 0 でないものが t 個あると仮定し，その番号 (誤り位置) i_r と値 e_{i_r} が未知であると考えてみましょう。改めて

$$x_r=\alpha^{n-i_r},\quad y_r=e_{i_r}\qquad(r=1,2,\cdots,t) \tag{27}$$

とおくと，式 (26) はこれら $2t$ 個の未知数に関する “非線形” 方程式

$$\begin{cases} y_1 & + & y_2 & +\cdots+ & y_t & = & s_1,\\ x_1\,y_1 & + & x_2\,y_2 & +\cdots+ & x_t\,y_t & = & s_2,\\ x_1^2\,y_1 & + & x_2^2\,y_2 & +\cdots+ & x_t^2\,y_t & = & s_3,\\ \quad\vdots & & \quad\vdots & & \quad\vdots & & \quad\vdots\\ x_1^{n-k-1}\,y_1 & + & x_2^{n-k-1}\,y_2 & +\cdots+ & x_t^{n-k-1}\,y_t & = & s_{n-k} \end{cases} \tag{28}$$

になり，未知数の個数 $2\,t$ が等式の個数 $n-k$ 以下 (すなわち, $t\le t_0$) ならば，解けることが期待されます。

実際，t が 1，および 2 のときは次のように解けます。$t=1$ の場合は簡単で，最初の 2 式

$$\begin{cases} y_1 & = & s_1,\\ x_1\,y_1 & = & s_2 \end{cases}\qquad(s_1\neq0\text{ を前提})$$

より $x_1=s_2/s_1$ で，その指数表示 $s_2/s_1=\alpha^{n-i_1}$ から誤り位置 i_1 が分かり，$e_{i_1}=y_1=s_1$ となります。

$t=2$ の場合は少し複雑で，一般の t に対する解法を示唆するので丁寧に見てみます。最初の 4 式は

$$\begin{cases} y_1+\ y_2 & = & s_1,\\ x_1\,y_1+x_2\,y_2 & = & s_2,\\ x_1^2\,y_1+x_2^2\,y_2 & = & s_3,\\ x_1^3\,y_1+x_2^3\,y_2 & = & s_4 \end{cases}\qquad(s_1\,s_3-s_2^2\neq0\text{ を前提})$$

です。第 1,2 式から

$$y_1=\frac{s_2-s_1\,x_2}{x_1-x_2},\qquad y_2=\frac{s_1\,x_1-s_2}{x_1-x_2} \tag{29}$$

を得て，これを第 3,4 式に代入すると

$$\begin{cases} s_2\,(x_1+x_2)-s_1\,x_1\,x_2 & = & s_3,\\ s_2\,(x_1^2+x_1\,x_2+x_2^2)-s_1\,x_1x_2\,(x_1+x_2) & = & s_4 \end{cases}$$

となります。これより和 x_1+x_2 と積 $x_1\,x_2$ が求まり，解と係数の関係より x_1, x_2 を解とする 2 次方程式

$$x^2-\frac{s_1\,s_4-s_2\,s_3}{s_1\,s_3-s_2^2}\,x+\frac{s_2\,s_4-s_3^2}{s_1\,s_3-s_2^2}=0\quad(s_1\,s_3-s_2^2\neq 0)\tag{30}$$

が得られます。方程式 (30) の解 $x_1=\alpha^{n-i_1}$, $x_2=\alpha^{n-i_2}$ は候補を次々に代入して見つけます。それを式 (29) に代入すれば誤り記号 $y_1=e_{i_1}$, $y_2=e_{i_2}$ の値が得られます。これで $t=2$ のときの問題も解決です。

この解法を一般化するために考察を続けます。方程式 (30) は形を整えると，

$$\begin{vmatrix} s_1 & s_2 \\ s_2 & s_3 \end{vmatrix} x^2-\begin{vmatrix} s_1 & s_3 \\ s_2 & s_4 \end{vmatrix} x+\begin{vmatrix} s_2 & s_3 \\ s_3 & s_4 \end{vmatrix}=0,$$

$$\text{ただし,}\begin{vmatrix} s_1 & s_2 \\ s_2 & s_3 \end{vmatrix}\neq 0,$$

したがって，

$$\begin{vmatrix} 1 & x & x^2 \\ s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \end{vmatrix}=0,\qquad \text{ただし, } |S|\neq 0\tag{31}$$

となります。式 (30) の左辺は 3.2 節で述べた誤り位置多項式 $\sigma(x)=x^2+\sigma_1\,x+\sigma_2$ で，係数 σ_1, σ_2 は連立 1 次方程式

$$\begin{bmatrix} s_1 & s_2 \\ s_2 & s_3 \end{bmatrix}\begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix}=\begin{bmatrix} s_3 \\ s_4 \end{bmatrix}$$

の解であって，クラーメルの公式 [7] の形になっています。したがって，掃き出し計算

$$\widetilde{S}=\begin{bmatrix} s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \end{bmatrix}\quad\longrightarrow\quad\begin{bmatrix} 1 & 0 & \sigma_2 \\ 0 & 1 & \sigma_1 \end{bmatrix}$$

により σ_1, σ_2 の値が得られ，2 次方程式 $x^2+\sigma_1\,x+\sigma_2=0$ を決定することができます。

以上の考察から，誤り個数が一般に $t\,(1\leq t\leq t_0)$ であるとき誤り位置 $x_r=\alpha^{n-i_r}\,(r=1,2,\cdots,t)$ を解とする方程式は，式 (31) から類推すると

$$\begin{vmatrix} 1 & x & \cdots & x^{t-1} & x^t \\ s_1 & s_2 & \cdots & s_t & s_{t+1} \\ s_2 & s_3 & \cdots & s_{t+1} & s_{t+2} \\ \vdots & \vdots & & \vdots & \vdots \\ s_t & s_{t+1} & \cdots & s_{2t-1} & s_{2t} \end{vmatrix}=0,\quad \text{ただし, } |S_t|\neq 0\tag{32}$$

[7] 方程式 $\begin{cases} a\,x+b\,y=p, \\ c\,x+d\,y=q \end{cases}$ の解は, $x=\dfrac{\Delta_1}{\Delta_0}$, $y=\dfrac{\Delta_2}{\Delta_0}$.

$\Delta_0=\begin{vmatrix} a & b \\ c & d \end{vmatrix}$, $\Delta_1=\begin{vmatrix} p & b \\ q & d \end{vmatrix}$, $\Delta_2=\begin{vmatrix} a & p \\ c & q \end{vmatrix}$.

ここでは記号 $\mathcal{A}$ の性質 $a=-a$ が使われる.

となりますが，実際，変数 x に x_r を代入すると行列式の値が 0 になるので [8]，この類推は正しいです。

この行列式を直接計算に使う必要はありません。方程式 (32) は $|S_t|\cdot\sigma(x)=0$ そのものであり，$|S_t|\neq 0$ ですから，誤り位置多項式 $\sigma(x)$ の係数 $\sigma_1,\cdots,\sigma_{t-1},\sigma_t$ は連立 1 次方程式

$$\begin{bmatrix} s_1 & s_2 & \cdots & s_t \\ s_2 & s_3 & \cdots & s_{t+1} \\ \vdots & \vdots & & \vdots \\ s_t & s_{t+1} & \cdots & s_{2t-1} \end{bmatrix}\begin{bmatrix} \sigma_t \\ \sigma_{t-1} \\ \vdots \\ \sigma_1 \end{bmatrix}=\begin{bmatrix} s_{t+1} \\ s_{t+2} \\ \vdots \\ s_{2t} \end{bmatrix}\tag{33}$$

の解となっています。その解を得るための掃き出し計算

$$\widetilde{S}_t=\begin{bmatrix} s_1 & s_2 & \cdots & s_t & s_{t+1} \\ s_2 & s_3 & \cdots & s_{t+1} & s_{t+2} \\ \vdots & \vdots & & \vdots & \vdots \\ s_t & s_{t+1} & \cdots & s_{2t-1} & s_{2t} \end{bmatrix}\longrightarrow\begin{bmatrix} 1 & 0 & \cdots & 0 & \sigma_t \\ 0 & 1 & \cdots & 0 & \sigma_{t-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & \sigma_1 \end{bmatrix}\tag{34}$$

が 3.3 節の掃き出し計算 (16) に当たります。そして，方程式

$$\sigma(x)=x^t+\sigma_1\,x^{t-1}+\cdots+\sigma_{t-1}\,x+\sigma_t=0\tag{35}$$

の変数 x に $\alpha^j\,(j=n-1,\cdots,1,0)$ を次々に代入していけば，解 $x_1,x_2,\cdots,x_t$ が見つかります。

x_r から誤り位置 i_r が分かるので，誤り記号 $y_r=e_{i_r}\,(r=1,2,\cdots,t)$ については，3.3 節に記した通り，式 (26) の前半 t 個の等式による連立 1 次方程式 (18) を解けば得られますから，課題 3° も解決です。

以上，$t=2$ の場合から類推して課題 2°，課題 3° の解決プロセスを導いてみました。

最後に，誤り個数 t (課題 1°) について考えます。性質 3 によると，$t\leq t_0$ である限り $t=\operatorname{rank}S$, $|S_t|\neq 0$ ですから，$\operatorname{rank}S$ を求めるために最初に掃き出し計算 (16) が必要でした。

$\operatorname{rank}S<t_0$ であれば $t=\operatorname{rank}S$ です。$\operatorname{rank}S=t_0$ のときは，3.3 節に記した通り，方程式 (35) の解が見つかる

[8] 例えば式 (31) の行列式の変数 x に x_1 を代入すると，

$$\begin{vmatrix} 1 & x_1 & x_1^2 \\ y_1+y_2 & x_1\,y_1+x_2\,y_2 & x_1^2\,y_1+x_2^2\,y_2 \\ x_1\,y_1+x_2\,y_2 & x_1^2\,y_1+x_2^2\,y_2 & x_1^3\,y_1+x_2^3\,y_2 \end{vmatrix}$$

$$=\begin{vmatrix} 1 & x_1 & x_1^2 \\ y_2 & x_2\,y_2 & x_2^2\,y_2 \\ x_2\,y_2 & x_2^2\,y_2 & x_2^3\,y_2 \end{vmatrix}=x_2\,y_2^2\begin{vmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_2 & x_2^2 \end{vmatrix}=0.$$

場合は $t = t_0$（訂正可能）で，見つからない場合は $t > t_0$（訂正不能）と判別できます。これで課題 1° も解決しました。

5.4　符号語間最小距離 $d_{\min}$ について

すでに 3.1 節で，(n, k) 短縮 RS 符号の訂正可能な誤り個数の限界 t_0 が $\lfloor (n-k)/2 \rfloor$ であることを述べ，その事実を使ってきました。ここで改めて式 (9) を証明します。

符号化 $G : \mathcal{A}^k \to \mathcal{A}^n$ は，k 次元空間にぎっしり詰まっている情報語を n 次元空間の中にばらまくことです。符号語をできるだけ分離して配置すれば訂正能力が増すことは，直感的に理解できます。

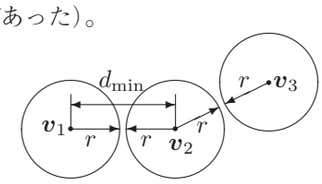
2 つの記号列間の**距離** D を記号列の中で対応する要素の異なり個数で計ることにします。すなわち，

$$\left.\begin{array}{l} \boldsymbol{a} = a_1\, a_2\, \cdots\, a_n, \\ \boldsymbol{b} = b_1\, b_2\, \cdots\, b_n \end{array}\right\} \text{に対して}$$
$$D(\boldsymbol{a}, \boldsymbol{b}) = \lceil a_i \neq b_i\,(1 \le i \le n) \text{ である } i \text{ の個数} \rfloor.$$

C の符号語間の最小距離を $d_{\min}$ で表します。そうすると訂正可能な誤り個数の限界 t_0 は

$$t_0 = \lfloor (d_{\min} - 1)/2 \rfloor \tag{36}$$

ということになります（図 2）。符号語 $\boldsymbol{v}$ から半径 t_0 の n 次元球を $\boldsymbol{v}$ の “勢力圏”と呼ぶと，異なる符号語の勢力圏は重なりません。しかし，n 次元空間を覆い尽くすことはできずに隙間が残ります（計算例で訂正不能の場合があった）。



球の半径 r は
$$t_0 = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$
に等しい。

図 2　符号語の “勢力圏”としての n 次元球

記号列 $\boldsymbol{w}$ の**重み** $W(\boldsymbol{w})$ を $\boldsymbol{w}$ のゼロでない記号の個数によって定義すると，2 つの符号語 $\boldsymbol{v}_1, \boldsymbol{v}_2$ に対して $D(\boldsymbol{v}_1, \boldsymbol{v}_2) = W(\boldsymbol{v}_1 - \boldsymbol{v}_2)$ です。線形符号では $\boldsymbol{v}_1 - \boldsymbol{v}_2$ も符号語ですから，次の性質が導かれます。

性質 4　線形符号における $d_{\min}$ は，$\boldsymbol{0}$ を除くすべての符号語の最小重みに等しい。すなわち，

$$d_{\min} = \min_{\boldsymbol{v}_1 \neq \boldsymbol{v}_2} D(\boldsymbol{v}_1, \boldsymbol{v}_2) \text{ とすれば, } d_{\min} = \min_{\boldsymbol{v} \neq \boldsymbol{0}} W(\boldsymbol{v}). \tag{37}$$

(n, k) 短縮 RS 符号における符号語間最小距離 $d_{\min}$ を考えるにあたり，検査行列 H に注目します：

$$H = \begin{bmatrix} 1 & \cdots & 1 & 1 \\ \alpha^{n-1} & \cdots & \alpha & 1 \\ \vdots & & \vdots & \vdots \\ \alpha^{(n-k-1)(n-1)} & \cdots & \alpha^{n-k-1} & 1 \end{bmatrix}.$$

これは性質 1 を満たすように生成多項式 $g(x)$ から式 (12) により定義された $(n-k) \times n$ 行列であって，第 $n-1$ 列に $g(x)$ の根 $1, \alpha, \alpha^2, \cdots, \alpha^{n-k-1}$ が並んでいます。α の指数が $0, 1, 2, \cdots, n-k-1$ と連続している ところに特徴があります。

この行列から任意に $n-k$ 個の列ベクトルを選び出した $n-k$ 次正方行列を考えます。列の番号を左から $j_1, j_2, \cdots, j_{n-k}$ として，簡単のため $a_r = \alpha^{n-j_r}$ $(r = 1, 2, \cdots, n-k)$ とおき，次のように表します。

$$H(j_1, j_2, \cdots, j_{n-k}) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ a_1 & a_2 & \cdots & a_{n-k} \\ \vdots & \vdots & & \vdots \\ a_1^{n-k-1} & a_2^{n-k-1} & \cdots & a_{n-k}^{n-k-1} \end{bmatrix}.$$

この行列の行列式は，すでに 5.1 節で述べたファンデルモンドの行列式ですから，$\prod_{i>j} (a_i - a_j)$ に等しく，a_r がすべて異なるのでその値は 0 にはなりません。そのことから次の性質が得られます。

性質 5　検査行列 H における任意の $n-k$ 個の列ベクトルは線形独立である。

「任意の $n-k$ 個」であることが重要です。これより次の性質 6 が導かれます。

まず，情報語 $\boldsymbol{u}$ が 1 個の要素を除きすべて 0 であるとき，組織符号化による符号語 $\boldsymbol{v} = \boldsymbol{u}\,G$ はその前部 $\boldsymbol{u}$ に $k-1$ 個の 0 を含むので $W(\boldsymbol{v}) \le n-k+1$ ですから，性質 4 より $d_{\min} \le n-k+1$ です。

行列 H^T の行ベクトルを $\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n$ とおくと，この中のどの $n-k$ 個も性質 5 により線形独立です。一方，任意の符号語 $\boldsymbol{v}\,(\neq \boldsymbol{0})$ に対して性質 2 により $\boldsymbol{v}\,H^{\mathrm{T}} = \sum_{j=1}^{n} v_j\,\boldsymbol{h}_j = \boldsymbol{0}$ ですから，$W(\boldsymbol{v}) \ge n-k+1$ でなければなりません。すなわち，$d_{\min} \ge n-k+1$ です。したがって，次の結論になります。

性質 6　(n, k) 短縮 RS 符号の符号語間最小距離 $d_{\min}$ は，

$$d_{\min} = n - k + 1. \tag{38}$$

実は線形とは限らない任意の (n, k) 符号に関して一般に，$d_{\min} \le n-k+1$ であることが知られています。(n, k) 短縮 RS 符号はこの限界を達成しているので，**最大距離分離符号**と呼ばれます。生成多項式の根が $1, \alpha, \cdots, \alpha^{n-k-1}$ で，その指数が連続している ことから性質 5，性質 6 が導かれたのでした[9]。

式 (36) と合わせると直ちに次の性質が得られ，式 (9) が証明されました。

[9] 生成多項式 $g(x)$ の根の指数が連続していないと線形独立性は保証されない．例えば，$n-k=3$ で，根が $1, \alpha, \alpha^3$ の場合，

$$\begin{vmatrix} 1 & 1 & 1 \\ a_1 & a_2 & a_3 \\ a_1^3 & a_2^3 & a_3^3 \end{vmatrix} = (a_1 + a_2 + a_3) \prod_{i>j} (a_i - a_j)$$

であるから，a_1, a_2, a_3 が互いに異なっても $a_1 + a_2 + a_3 = 0$ という関係にある場合，この 3 個の列ベクトルは線形従属となる．

性質 7　(n,k) 短縮 RS 符号において，訂正可能な誤り個数の限界 t_0 は，

$$t_0 = \left\lfloor \frac{n-k}{2} \right\rfloor . \tag{39}$$

3 章で紹介した符号化・復号アルゴリズムが，誤り個数 t が t_0 以下のときに正しく働くことを示す長いストーリーは，これで終わりです。

6　一般と特殊

QR コードの一部には 2 元 BCH 符号と呼ばれる符号が使われているので，それを簡単に紹介します。線形符号に巡回符号と呼ばれるクラスがあり，BCH 符号や RS 符号はこのクラスに属します。短縮 RS 符号の周辺の各種の符号の関係を「一般と特殊」という視点で整理してみます。

6.1　符号語長 n が $q-1$ に等しい場合（巡回符号）

(n,k) 短縮 RS 符号はその符号語長 n が $q-1$ に等しいとき，“正真正銘”の **RS 符号**となります。RS 符号では符号語長が固定されて不便のため，それを自由な長さに短縮したものが短縮 RS 符号です。

一般に，(n,k) 線形符号 C の組織符号化において，左端 ν 個 $(\nu < k)$ がすべて 0 の情報語に限定すると，対応する符号語も左端 ν 個が 0 となる部分符号が得られ，余分な 0 を除去した $(n-\nu, k-\nu)$ 符号も線形で，これを C の**短縮符号** (shortened code) と呼びます。短縮しても符号語間の最小距離は減少しません。

RS 符号とは，加減乗除の定義された q 個の要素からなる記号系 $\mathcal{A} = \{0, 1, \alpha, \cdots, \alpha^{q-2}\}$ において，指数の連続する h 個の根 $\alpha^{\ell}, \alpha^{\ell+1}, \cdots, \alpha^{\ell+h-1}$ をもつ生成多項式 $g(x)$ によって定義される $(q-1, q-1-h)$ 符号です。この解説では $q = 2^8$, $h = q-1-k$, $\ell = 0$ としてきました。

$n = q-1$ である RS 符号は，$\alpha^n = 1$ であることから任意の符号語 $\boldsymbol{v} = v_1\, v_2 \cdots v_n$ の巡回シフト記号列 $\boldsymbol{v}' = v_2\, v_3 \cdots v_n\, v_1$ も符号語になります。この性質をもつ符号は**巡回符号** (cyclic code) と呼ばれ，装置化が容易なことから応用上重要な符号のクラスです (短縮すると巡回性を失う)。

一般に，(n,k) 巡回符号は $x^n - 1$ を割り切る生成多項式 $g(x)$ をもち，検査多項式 $h(x) = (x^n-1)/g(x)$ が定義されます。n 個の記号 $1, \alpha, \alpha^2, \cdots, \alpha^{n-1}$ は方程式 $x^n - 1 = 0$ の相異なる根になるので，$g(x)$ はこの中の $n-k$ 個を根にもち，$h(x)$ は残りの k 個を根にもちます。RS 符号では生成多項式の根の指数が連続していることで，誤り訂正能力に優れます（性質 7)。

巡回符号ではないが生成多項式をもつ線形符号のクラスは**擬巡回符号**（pseudo-cyclic code, 巡回符号に準ずるの意）と呼ばれ，短縮 RS 符号はこのクラスに属します。

6.2　さらなる拡張 (BCH 符号)

q 個の要素の符号アルファベット $\mathcal{A}$ で構成される符号を q 元符号と呼びます。q 元巡回符号の生成多項式の根を，$\mathcal{A}$ をさらに拡大した要素数 $q^{m'}$ の記号系 $\mathcal{B}$ の中で考えることができます。$\mathcal{B}$ の 1 つの要素を $\beta \neq 0$，$\beta^{n'} = 1$（n' は $q^{m'}-1$ の約数）とします。根として $\beta^{\ell}, \beta^{\ell+1}, \cdots, \beta^{\ell+h-1}$ をすべて含み，係数がすべて $\mathcal{A}$ の要素である多項式 $g(x)$ を考えます。この $g(x)$ を生成多項式にもつ長さ n' の q 元符号を **BCH 符号** (Bose-Chaudhuri-Hocquenghem code) と呼びます。これは RS 符号を一般化したもので，とくに $m' = 1, \mathcal{B} = \mathcal{A}, n' = q-1, \beta = \alpha$ である BCH 符号が RS 符号ということになります。

β^j と他の根を組み合わせて係数が $\mathcal{A}$ の要素となるような最小次数の多項式を β^j の最小多項式と呼んで $M_j(x)$ で表し，多項式 $p_1(x), p_2(x)$ の最小公倍多項式を $\mathrm{LCM}[\, p_1(x), p_2(x)\,]$ で表すことにすれば，今示した BCH 符号の生成多項式は

$$g(x) = \mathrm{LCM}\,[\, M_{\ell}(x),\, M_{\ell+1}(x),\, \cdots,\, M_{\ell+h-1}(x)\,]$$

と表現されます。β^j と β^{2j} の最小多項式は同一なので，j が奇数の最小多項式 $M_j(x)$ だけの最小公倍多項式を考えればすみます。多項式 $g(x)$ の次数を d とすると，$(n', n'-d)$ 線形符号が生成されます。

とくに $q = 2$，$\mathcal{A} = \{0, 1\}$ である **2 元 BCH 符号**がよく使われます。QR コードでは形式情報 (5 ビット) を $(15, 5)$ 2 元 BCH 符号，型番情報 (6 ビット) を $(18, 6)$ 2 元 BCH 符号によって表しています。

例 1　$q = 2, m' = 4$, $n = n' = 15$ とし，β を $\beta^4 + \beta + 1 = 0$ $(\beta^{15} = 1)$ の関係を満たす記号とします（付録 1 参照)。$\ell = 1$, $h = 6$ とし，$\beta, \cdots, \beta^6$ を根とする 2 元 BCH 符号の最小多項式を $g(x)$ とします。β, β^3, β^5 の最小多項式はそれぞれ $M_1(x) = (x-\beta)(x-\beta^2)(x-\beta^4)(x-\beta^8) = x^4 + x + 1$, $M_3(x) = (x-\beta^3)(x-\beta^6)(x-\beta^{12})(x-\beta^9) = x^4 + x^3 + x^2 + x + 1$, $M_5(x) = (x-\beta^5)(x-\beta^{10}) = x^2 + x + 1$ であり，これらは共通の根を持たないので，生成多項式は

$$g(x) = M_1(x)\, M_3(x)\, M_5(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$$

となります。$g(x)$ の次数は 10 ですから $k = n - 10 = 5$ であり，係数列 $\boldsymbol{g} = 1\,0\,1\,0\,0\,1\,1\,0\,1\,1\,1$ から生成行列 G_0 と組織符号のための生成行列 G は次のようになります。

$$G_0 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix},$$

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

この $(15,5)$ 符号の誤り訂正能力は $t_0=\lfloor h/2\rfloor=3$ です。

QR コードでは 4 種の誤り訂正レベル L(7%), M(15%), Q(25%), H(30%) の選択と 8 種のマスクパターンの選択をそれぞれ 2 ビット，3 ビットで表現し，この 5 ビットを形式情報と呼びます。形式情報を表す情報語を例えば $\boldsymbol{u}=11011$, $u(x)=x^4+x^3+x+1$ とすると，$u(x)\,x^{10}$ を $g(x)$ で割った余り $r(x)=x^9+x^4+x^2$ から符号多項式 $v(x)=u(x)\,x^{10}-r(x)$ が，また G により符号語 $\boldsymbol{v}=110111000010100$ が得られます。受信語の復号（誤り訂正）の方法は短縮 RS 符号と同様ですが，この場合は符号語の総数が $2^5=32$ と少ないので，受信語と最短距離にある符号語を全探索して得る復号法も使うことができます。

例 2　$\ell=1$, $h=2$ の 2 元 BCH 符号は，β, β^2 を根とする生成多項式

$$g(x)=M_1(x)=\prod_{j=0}^{m-1}(x-\beta^{2^j})$$

により定義される $(2^m-1,\ 2^m-1-m)$ 符号は**巡回ハミング符号** (cyclic Hamming code) と呼ばれ，m によらず $t_0=1$ です。例えば $m=3$ のときは $g(x)=x^3+x+1$, $h(x)=x^4+x^2+x+1$ となり，生成行列 G_0 と検査行列 H_0 は以下のようになります。

$$G_0=\begin{bmatrix} 1&0&1&1&0&0&0\\ 0&1&0&1&1&0&0\\ 0&0&1&0&1&1&0\\ 0&0&0&1&0&1&1 \end{bmatrix},\ H_0=\begin{bmatrix} 1&1&1&0&1&0&0\\ 0&1&1&1&0&1&0\\ 0&0&1&1&1&0&1 \end{bmatrix}.$$

参考までに巡回性をもたない (7,4) ハミング符号[10] の生成行列 G_1 を以下に示します。

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

各種の線形符号の関係を図に表すと，この解説が対象としてきた短縮 RS 符号の位置がよくわかります。

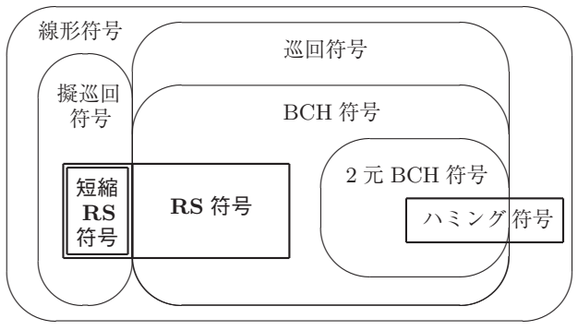


図 4　各種線形符号の関係

あとがき

QR コードは株式会社日本電装（現在，デンソーウェーブとして独立）によって開発され，1994 年に発表された 2 次元バーコードの呼称（Quick Response の略）で，高速読み取りができる特性をアピールしています。1999 年に JIS X0510 という規格になり，さらに 2000 年には国際規格 ISO/IEC 18004 として認定されました。QR コードは部品や商品の管理情報を記録するために開発されましたが，現在ではカメラ付き携帯電話に復号機能が内蔵され，Web サイトの URL をはじめ多種多様の情報を手軽に伝達する手段として急速に普及してきました。

QR コードにおいて誤り訂正のために用いる RS 符号は，CD や DVD などの記録方式，ADSL や衛星放送などの通信方式において標準的に利用されている符号ですが，50 年前に発明されたものです[11]。その原理は決してすぐに分かるものではありませんが，「なぜうまくいくのだろう？!」と知的好奇心を膨らませながら，少しずつ理解を進めてほしいと思います。仕組みが分かってくるとその巧妙さに感激し，ものごとを深く理解する喜びを十分に味わうことができます。

冒頭に，行列やベクトルの知識を前提にすると記しましたが，実際にこの世界でも「線形代数」が縦横に活躍しました。その底力には改めて感服するとともに，基礎教育の大切さを思い知らされます。

逆に言うと，線形代数を学んでないとこの解説を理解するのは少し難しいと思いますが，目標をもって教科書を開けば，自ら学ぶ喜びが得られることでしょう。

この解説で紹介したアルゴリズムは**ピーターソン法** (Peterson algorithm) と呼ばれるものです。携帯電話に組み込まれる方法は，処理の高速化のためにさらに改良されているでしょう。この解説が，読者諸氏の興味をさらに深めるきっかけになることを願っています。

草稿を読んでいただき、いくつかの誤りを指摘して下さった秋草学園短期大学の斉藤奈保子教授に感謝します。

[10] 検査行列の列にゼロベクトルを除くすべてのベクトルが並ぶような線形符号を**ハミング符号**（Hamming code）と呼ぶ.

[11] 2010 年 9 月の電子情報通信学会では「リード・ソロモン符号 50 年」と題する記念講演会が催された.

付録

1. 加減乗除のできる有限の記号系 $\mathcal{A}$

$\mathcal{A}$ が $q = 2^8$ 個の要素をもつ場合，多項式 $P(x) = x^8 + x^4 + x^3 + x^2 + 1$ を選び（その理由は説明外），関係 $P(\alpha) = 0$ を満たす記号 α を使って，$\mathcal{A} = \{0, 1, \alpha, \alpha^2, \cdots, \alpha^{254}\}$ と表した。$\mathcal{A}$ のすべての記号は，7次以下の α の多項式 $b_1\,\alpha^7 + b_2\,\alpha^6 + \cdots + b_7\,\alpha + b_8$ (係数 b_i は 0 か 1) で表されるので，指数表現 α^j は長さ 8 のビット列表現 $b_1\,b_2\cdots b_8$ と 1 対 1 に対応する。

例えば，$\alpha^8 = \alpha^4 + \alpha^3 + \alpha^2 + 1 \leftrightarrow 00011101$, $\alpha^9 = \alpha^5 + \alpha^4 + \alpha^3 + \alpha \leftrightarrow 00111010$ である。

指数表現とビット列表現の対応表を得るには，$\alpha^j = \sum_{i=1}^{8} b_i\,\alpha^{8-i}$ のとき $\alpha^{j+1} = b_1\,\alpha^8 + \sum_{i=2}^{8} b_i\,\alpha^{9-i} = b_2\,\alpha^7 + b_3\,\alpha^6 + b_4\,\alpha^5 + (b_5 + b_1)\,\alpha^4 + (b_6 + b_1)\,\alpha^3 + (b_7 + b_1)\,\alpha^2 + b_8\,\alpha + b_1$ という関係を漸化的に用いる（$\alpha^0 \leftrightarrow 00000001$ から開始する）。

この関係 $\alpha^j \Rightarrow \alpha^{j+1}$ は下図の 8 段フィードバック・シフトレジスタと呼ばれる図式によって表現できる。

乗除算に関しては本文に述べた通りとし，加算 (減算も同じ) $\alpha^i + \alpha^j$ は次のように定義する。α^i, α^j をそれぞれビット列で表現し，それら 2 つのビット列の和をビット毎の排他的論理和（XOR）によって生成し，そのビット列が α^ℓ のビット列表現に等しいとき，$\alpha^i + \alpha^j = \alpha^\ell$ と定義する。

例えば，$\alpha^8 + \alpha^9 \leftrightarrow 00011101 + 00111010 = 00100111$ であり，一方 $\alpha^{33} = \alpha^5 + \alpha^2 + \alpha + 1 \leftrightarrow 00100111$ であるから，$\alpha^8 + \alpha^9 = \alpha^{33}$ である。

$$\begin{array}{r} 00011101\ (\leftarrow \alpha^8) \\ +\ 00111010\ (\leftarrow \alpha^9) \\ \hline 00100111\ (\rightarrow \alpha^{33}) \end{array}$$

指数表現とビット列表現の対応表を作れば，加減算の実行が速やかになる。次の表では便宜上，0 の指数を -1 で表した。$e(b(j)) = j$, $b(e(j)) = j$ である。

j	α^j のビット列表現	10 進 $b(j)$	逆引き $e(j)$
-1	00000000	0	—
0	00000001	1	-1
1	00000010	2	0
2	00000100	4	1
3	00001000	8	25
4	00010000	16	2
5	00100000	32	50
6	01000000	64	26
7	10000000	128	198
8	00011101	29	3
9	00111010	58	223
10	01110100	116	51

加算表 $\alpha^i + \alpha^j$ を作ればもっと早くなるが，表の大きさは 256×256 になる。その一部を示す。

	-1	0	1	2	3	4	5	6	7	8	9	10
-1	-1	0	1	2	3	4	5	6	7	8	9	10
0	0	-1	25	50	223	100	138	191	112	200	120	21
1	1	25	-1	26	51	224	101	139	192	113	201	121
2	2	50	26	-1	27	52	225	102	140	193	114	202
3	3	223	51	27	-1	28	53	226	103	141	194	115
4	4	100	224	52	28	-1	29	54	227	104	142	195
5	5	138	101	225	53	29	-1	30	55	228	105	143
6	6	191	139	102	226	54	30	-1	31	56	229	106
7	7	112	192	140	103	227	55	31	-1	32	57	230
8	8	200	113	193	141	104	228	56	32	-1	33	58
9	9	120	201	114	194	142	105	229	57	33	-1	34
10	10	21	121	202	115	195	143	106	230	58	34	-1

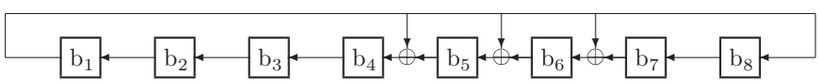
2. 短縮 RS 符号の生成多項式

QR コードで使われる r 次の生成多項式 $g(x)$（本文の式 (2)）の展開係数を高次順に指数で示す（一部のみ）。

例えば，$r = 7$ の (87, 229, 146, 149, 238, 102, 21) は $g(x) = x^7 + \alpha^{87}x^6 + \cdots + \alpha^{102}x + \alpha^{21}$ を表す。

r	r 次の生成多項式 $g(x)$ の係数
2	(25, 1)
3	(198, 199, 3)
4	(75, 249, 78, 6)
5	(113, 164, 166, 119, 10)
6	(166, 0, 134, 5, 176, 15)
7	(87, 229, 146, 149, 238, 102, 21)
8	(175, 238, 208, 249, 215, 252, 196, 28)
9	(95, 246, 137, 231, 235, 149, 11, 123, 36)
10	(251, 67, 46, 61, 118, 70, 64, 94, 32, 45)
11	(220, 192, 91, 194, 172, 177, 209, 116, 227, 10, 55)
12	(102, 43, 98, 121, 187, 113, 198, 143, 131, 87, 157, 66)
13	(74, 152, 176, 100, 86, 100, 106, 104, 130, 218, 206, 140, 78)
14	(199, 249, 155, 48, 190, 124, 218, 137, 216, 87, 207, 59, 22, 91)
15	(8, 183, 61, 91, 202, 37, 51, 58, 58, 237, 140, 124, 5, 99, 105)
16	(120, 104, 107, 109, 102, 161, 76, 3, 91, 191, 147, 169, 182, 194, 225, 120)
17	(43, 139, 206, 78, 43, 239, 123, 206, 214, 147, 24, 99, 150, 39, 243, 163, 136)

例えば，QR コード型番 1 では訂正レベルに応じて次数 $r = 7, 10, 13, 17$ の $g(x)$ を使う。JIS 規格 X 510 に必要な $g(x)$ の表があるが，定義から計算可能。r 次の生成多項式を $g_r(x)$ とすれば，$g_{r+1}(x) = g_r(x)(x - \alpha^r)$ である。



8 段フィードバック・シフトレジスタ （⊕ は XOR を表す）

3. 符号化・復号プログラム

参考までに, 生成多項式 $g(x)$ の (n, k) 短縮 RS 符号の符号化と復号を Mathematica プログラムで示す。

記号系 $\mathcal{A}$ の要素は指数表現 α^j の j で（0 の指数を便宜上 -1 で）表す。予め，$\mathcal{A}$ における加減算 add，乗算 mul，逆数 inv，累乗 pow，ベクトルの内積 sekiwa，多項式の値の計算 evalpoly，行列の掃き出し計算 sweepout（階数，連立 1 次方程式の解，標準形を出力）などの関数（プログラムは省略）を用意する。

n，k，nk $(= n - k)$，gmat，hmat は大域的変数（gmat，hmat は符号化・復号の前に準備する）。

大文字で始まる Module，Table，Transpose などの関数は Mathematica の関数である。

- **符号化関数** encoder への入力は情報語 u，出力は符号語 v である。

```
encoder[u_] := Module[{r, i, v},
  r = Table[sekiwa[u, Transpose[gmat][[i]]],{i, k+1, n}]; (* ベクトル表現 *)
  v = Join[u, r];
  Return[v]
];
```

r は多項式 $u(x)\,x^{n-k}$ を $g(x)$ で割った余り $r(x)$ の係数列 $\boldsymbol{r}$ に対応し，$\boldsymbol{r} = \boldsymbol{u}\,P$ により計算する。

生成行列 gmat から単位行列を除いた $k \times (n-k)$ 部分行列が P になる。

- **復号関数** decoder への入力は受信語 w，出力は情報語 uu である。

```
decoder[w_] :=
Module[{uu, t0, t, i, j, s, zero, smat, sig, sig0, emat, e, errloc, errval},
  s = Table[sekiwa[w,hmat[[i]]],{i,nk}]; (* シンドロームの計算 ベクトル表現 *)
  zero = Table[-1, {nk}];
  If[s == zero, Print["誤りなし"]; uu = Take[w, k]; Return[uu]];
  t0 = Floor[nk/2];   (* t0 は訂正可能な誤り数 *)
  smat = Table[s[[i+j-1]], {i, t0}, {j, t0+1}];(* smat は拡大シンドローム行列 *)
  sig = sweepout[smat][[2]]; (* smat の掃き出し計算 *)
  sig0 = Prepend[Reverse[sig], 0]; (* sig0 は誤り位置多項式 σ(x) の係数列 *)
  errloc = {};
  Do[If[evalpoly[sig0, j] == -1, errloc = Append[errloc, n-j]], {j, n-1, 0, -1}];
  t = Length[errloc]; (* errloc は誤り位置，t は誤り個数 *)
  If[t == 0, Print["訂正不能！"]; uu = Take[w, k]; Return[uu]];
  emat = Table[Append[Table[pow[i-1, n-errloc[[j]]], {j, t}], s[[i]]], {i, t}];
  errval = sweepout[emat][[2]]; (* 誤り記号 errval の計算 *)
  e = Table[-1, {n}];
  Do[i = errloc[[j]]; e[[i]] = errval[[j]], {j, t}]; (* e は誤り語 *)
  uu = Table[add[w[[i]], e[[i]]], {i, k}];
  Return[uu]
];
```

s はシンドローム・ベクトルで，受信語 w と検査行列 hmat との積（積和の反復）によって計算する。

t0 は本文における t_0，smat は本文における拡大シンドローム行列 $\widetilde{S}$ である。

4. QR コードの概要

さまざまなビット列，数列，文字列を実際に QR コード・シンボルで表すためには，詳細な作成仕様を JIS 規格 X 0510 で知る必要がある。ここにはその概要を記すにとどめる。

4.1 構成単位と形状

QR コード・シンボルの構成単位は正方形，ときに円形のセル (JIS ではモジュールと呼ぶ) で明暗（白黒）の 2 値をとる。

セルは正方マトリックス状に配置され，1 辺のセル数は 1〜40 の型番により 21 から 177 まで 4 ずつ増える（型番 a の 1 辺のセル数は $4\,a + 17$ である）。

QR コード・シンボルは，クワイエット・ゾーンと称する明領域 (幅 4 セル) で囲まれているものとする。

4.2 固定領域

QR コード・シンボルの範囲と方向を定めるために設定されるパターンで，型番ごとに固定している。それらは，(1) 位置検出パターン（同心明暗正方形 7×7 セル，3 ヶ所），(2) 分離パターン（位置検出パターン 2 辺を囲む L 字型明帯 15 セル，3 個），(3) タイミングパターン（位置検出パターンを繋ぐ明暗帯，縦横各 1 本），(4) 位置合せパターン（型番 2 以上，同心明暗正方形 5×5 セル，1〜46 個），(5) 左下の位置検出パターン角の暗セル 1 個，からなる。なお，左右上下の方向は位置検出パターンのないコーナーが右下となる基準で記す。

型番 a の固定領域のセル数 b は，$b = 8a + 139 + (5\lfloor a/7 \rfloor + 9)^2$（例外は $a = 1$ のとき $b = 202$）である。

4.3 符号化領域

表現される情報により変化する可変領域で，(1) 形式情報（訂正レベルとマスクパターン指示子，15 セル×2），(2) 型番情報（型番 7 以上，18 セル×2，位置検出パターン右上の左部と左下の上部），(3) データ領域（符号語部分），(4) 残余（0〜7 セル）からなる。

型番 a の符号化領域のセル数 c は，$c = (4a+17)^2 - b$ であり，その中で形式情報部が 30 セル，$a \geq 7$ のとき型番情報部が 36 セルを占める。データ領域のバイト数 d は，$d = 2(a+4)^2 - 19 - 3(\lfloor a/7 \rfloor + 2)^2$ で表される（ただし，$2 \leq a \leq 6$ のときは 3 を加え，$a = 1$ のときは $d = 26$）。データ領域の割合は，型番 1 で 47%，型番 2 で 56%, 以下次第に増加して型番 40 では 95%になる。

4.4 データ領域の生成

データ領域に符号化される情報は 1 個以上のセグメントからなり，各セグメントは同じモード（文字種）のビット列で表現され，モード指示子と文字数指示子が先行し，必要に応じて「埋め草」，終端パターンが付加される。8 ビットを $\mathcal{A}$ の 1 記号に対応させて，JIS に定められたパラメータ（一般には 2 種類の）(n, k) の短縮 RS 符号化を一定回数繰り返す。1 個の符号語の領域 $8n$ セルをブロックと呼ぶ。例えば型番 5 で訂正レベル Q の場合，$(33, 15)$ 符号化と $(34, 16)$ 符号化がそれぞれ 2 回繰り返され，4 ブロックとなる。複数のブロックからなる場合，各ブロックから 1 記号ずつ巡回的に取り出して並べ直す。

並べ直しの例（型番 5 訂正レベル Q）

ブロック	符号語	情報部	冗長部
1	$\boldsymbol{v}_1$	$u_{1,1} \cdots u_{1,15}$	$r_{1,1} \cdots r_{1,18}$
2	$\boldsymbol{v}_2$	$u_{2,1} \cdots u_{2,15}$	$r_{2,1} \cdots r_{2,18}$
3	$\boldsymbol{v}_3$	$u_{3,1} \cdots u_{3,15}\ u_{3,16}$	$r_{3,1} \cdots r_{3,18}$
4	$\boldsymbol{v}_4$	$u_{4,1} \cdots u_{4,15}\ u_{4,16}$	$r_{4,1} \cdots r_{4,18}$

$$u_{1,1}\, u_{2,1} \cdots u_{3,15}\, u_{4,15}\, \underline{u_{3,16}\, u_{4,16}}\, r_{1,1}\, r_{2,1} \cdots r_{3,18}\, r_{4,18}$$

並べ直した記号列をビット列で表現した後，データ領域に配置する。その配置方法は次の通りである (JIS X 0510 の “ シンボルにおける配置の別法 ” 参照)：

データ領域を可能な限り 2 列単位に分割する。右下のセルから開始して 2 列単位にジグザクに上下する方向で進む。進みながら同じ行では右を優先させて上位ビットから下位ビットの順に配置する。すべての符号語が終わるまで繰り返し，残余領域があれば 0 で埋める。固定された除外領域のため，1 記号に対応する 8 セルは必ずしも 2×4 の長方形状にはならない（連結しないこともある）。

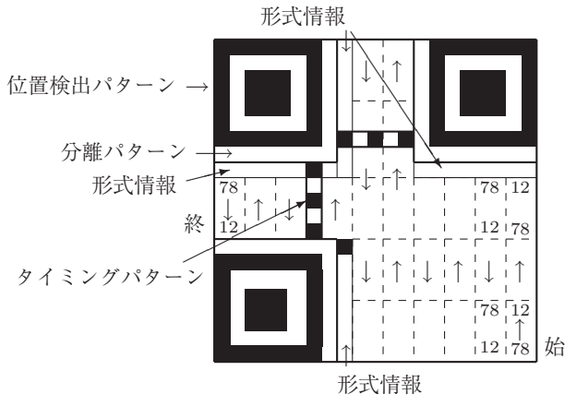
最終的な明暗パターンは 8 種類中で最適なマスクパターンとの排他的論理和によって生成される。“最適性” は，明や暗の連続が少ないこと，位置検出パターンの走査パターンと同じものが少ないことなどで評価する。

4.5 訂正レベルについて

各型番で 4 種の誤り訂正レベル L(7%), M(15%), Q(25%), H(30%) を選択できるが, この%の数字は訂正可能な損傷セルの比率を意味しない。(n, k) 符号化を a 回，$(n+1, k+1)$ の符号化を b 回使用する訂正レベルは，符号化領域における損傷記号数（8 セル単位）の割合

$$\rho = \frac{\lfloor \frac{n-k}{2} \rfloor}{a+b} \left(\frac{a}{n} + \frac{b}{n+1} \right) \approx \frac{n-k}{2n+1}$$

を目安に設定されたものと考えられる。例えば型番 5 で訂正レベル Q の場合，$n = 33, k = 15, a = b = 2$ であるから $\rho = 26.87\%$ である。面積比ではなく，固定領域の損傷も考慮外であるので，実際の感覚とは異なる。



87⋯21 はビット列表現 $b_8\, b_7 \cdots b_2\, b_1$ に対応 (b_8 が上位ビットを表す)
型番 1 では 1 記号に対応する 8 セルがすべて長方形状（一部非連結）に配置される．

- 型番 2 より「位置合せパターン」がある．
- 型番 7 より「型番情報」が入る．

QR コードの構造　（最小の型番 1 を例として）

5. QR コードにおける漢字の表現

QR コードでは漢字かな混じり文は冗長性をなくして 1 文字 13 ビットで表現する。例えば，シフト JIS コードとの間に次のような変換が必要になる。シフト JIS コードの上位バイト，下位バイトの対を (x, y) とし，図のように平面上に配置すると，$[81] \le x \le [9f]$ と $[e0] \le x \le [ea]$ の 2 つのエリアに分かれ，$[40] \le y \le [fc]$ の範囲にある（$[\cdot]$ は 16 進数を表す）。

まず，2 つのエリアを左上に平行移動して合併すると，

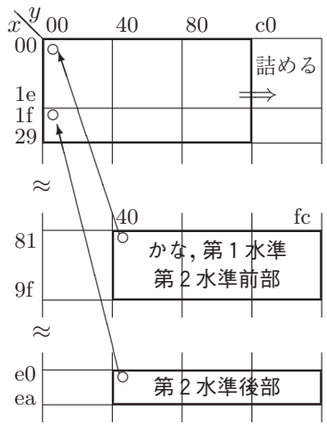
$$(x-d,\ y-[40]),$$
$$\text{ただし，}\ d = \begin{cases} [81], & x \le [9f] \text{ のとき} \\ [c1], & x \ge [e0] \text{ のとき} \end{cases}$$

となる。さらに，下位バイトが [c0] 以上の部分に詰めて 1 列に並べ，

$$z = (x-d) \times [c0] + (y-[40])$$

とおけば $0 \le z < 2^{13}$ となるので，z は 13 ビットで表現できる。これが QR コードにおける漢字コードである。逆変換は次の通り。

$$x = \left\lfloor \frac{z}{[c0]} \right\rfloor + d, \qquad y = \bmod (z, [c0]) + [40].$$



例　「春は名のみ」

→　[8f74], [82cd], [96bc], [82cc], [82dd]（シフト JIS）

→　[0ab4], [014d], [103c], [014c], [015d]（13 ビット）

参考文献

（発行の新しい順）

[1] 濱屋　進,「符号理論入門」，工学社，2008.
[2] 池田和興,「例題が語る符号理論 – BCH 符号・RS 符号・QR コード」，共立出版，2007.
[3] J. ユステセン, T. ホーホルト著（坂田省二郎ほか訳）,「誤り訂正符号入門」，森北出版，2005.
[4] 情報理論とその応用学会編,「符号理論とその応用」，培風館，2003.
[5] 標準化研究学会編,「QR コードのおはなし」，日本規格協会，2002.
[6] 平澤茂一, 西島利尚,「符号理論入門」，情報数理シリーズ A-3，培風館，1999.
[7] 映像情報メディア学会編,「誤り訂正符号とその応用」，オーム社，1996.
[8] 今井秀樹,「符号理論」，電子情報通信学会，1990.
[9] 蓮井　敏,「Handbook 線形代数学」，共立出版，1987.