セグメントが誤り訂正符号となる周期数列の代数的構成とＣ言語による探索

1271040B
夏　文雄

2004年2月10日

1 error correcting code

定義 1.1. X:有限集合, $\#(X)=q$ とする。X 上の n 次 $DeBruijn$ 系列とは周期 q^n の X 上の数列であってサイズが n の window property をもつもの。

例 $0001011100010111\cdots$ は $X=\mathbf{F}_2=\{0,1\}$ 上の 3 次の $DeBruijn$ 系列である。

定理 1.3. の証明の準備として次のことを示しておく。

定理 1.2(Euler の定理) 強連結グラフの任意の頂点において入り次数と出次数が等しいならば, そのグラフは Euler 回路である。

定理 1.3. 任意の $q\in\mathbf{N}$, 任意の $n\in\mathbf{N}$ に対して $DeBruijn$ 系列はある。

証明 $(\mathbf{F}_q)^{n-1}$の全ての元を頂点とし, 頂点 x が表すベクトルを 1 ビットシフトして空いた最下位ビットに $\mathbf{F}_q$の元を入れて, それが頂点 y が表すベクトルになるとき x から y に有向辺を結ぶグラフを考える。(このグラフを $DeBruijn$ グラフという。) このグラフが一筆書き可能なことと, $\mathbf{F}_q$上の n 次の $DeBruijn$ 系列が存在することとは同値である。
定理 1.2.を用いて $DeBruijn$ グラフが一筆書き可能なことを示す。任意の頂点 x,y に対して x から y に行く有向道は存在する。なぜなら $n-1$ ビットのベクトルなので $n-1$ 回以下のシフトで必ず目的のベクトルに出来る。つまりグラフは強連結である。
任意の頂点 x に対して (入り次数) = (出次数) $=q$ である。なぜなら x をシフトして $\mathbf{F}_q$の元を入れるということは x の出次数は q である。y をシフトして x になるということは y の下 $n-2$ ビットと x の上から $n-2$ ビットが一致している。そのような y は最上位ビットの選び方が q 通りあることから q 個である。つまり x の (入り次数) $=q$。
以上のことで定理 1.2.を適用するとこのグラフはオイラー回路である。つまり一筆書き可能である。よって $DeBruijn$ 系列は存在する。

$DeBruijn$ 系列の個数に関して次のことが知られている。

定理 1.4. $q=2$ のときの n 次の DeBruijn 系列の個数は

$$2^{2^{n-1}-n}$$

である。(ただし $00\cdots001$ からスタートしたときの個数)

距離関数 関数　$d : (\mathbf{F}_q)^m \times (\mathbf{F}_q)^m \longrightarrow \mathbf{N}$　を次のように定義する。

$$d(\mathbf{x}, \mathbf{y}) \vcentcolon = d((x_1, x_2, \cdots, x_m), (y_1, y_2, \cdots, y_m)) = (x_i \neq y_i \text{となる } i \text{ の個数})$$

これは $(\mathbf{F}_q)^m$ 上の距離関数となる。($Hamming$ 距離という)

証明 $(1) d(\mathbf{x}, \mathbf{y}) \geq 0 \ \ \forall \mathbf{x}, \mathbf{y} \in (\mathbf{F}_q)^m, \ d(\mathbf{x}, \mathbf{y}) = 0 \Longleftrightarrow \mathbf{x} = \mathbf{y}$
$(2) d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}) \ \ \forall \mathbf{x}, \mathbf{y} \in (\mathbf{F}_q)^m$
この 2 つは明らか。
$(3) d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y}) \quad \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in (\mathbf{F}_q)^m$ を示す。
この式を定義に戻り書き直すと
$(x_i \neq y_i \text{となる } i \text{ の個数}) \leq (x_i \neq z_i \text{となる } i \text{ の個数}) + (z_i \neq y_i \text{となる } i \text{ の個数})$ となる。
$x_k \neq y_k \Longrightarrow \bullet x_k \neq z_k, y_k \neq z_k$
$\bullet \ x_k = z_k, y_k \neq z_k$
$\bullet \ x_k \neq z_k, y_k = z_k$
の 3 つの場合があるが, 左辺はこの k に関して 1 回カウントするが, 右辺はこの k に関して 1 回以上カウントする。従って (3) も成り立つ。

以下距離関数 d は $Hamming$ 距離とする。

定義 1.5.　上で定義した距離に関して $C \subset (\mathbf{F}_q)^m$ の元を中心とする半径 e の球が交わらないとき, C を符号長 m の e-error correcting code(e 誤り訂正符号) という。

e-error correcting code は符号の送受信中に e ビット以内の誤りであれば正確にもとの符号に復号することが出来る符号である。e をこの符号の誤り訂正能力という。

補題 1.6. C が e-error correcting code であるための必要十分条件は,
$$min\ dist\ of\ C \vcentcolon = min\{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \neq \mathbf{y}, \mathbf{x}, \mathbf{y} \in C\} \geq 2e + 1$$
となることである。

証明 $B(\mathbf{x}, e)$ を $\mathbf{x} \in C$ を中心とする半径 e の円とする。
$\Longrightarrow$ を示す。任意に $\mathbf{x}, \mathbf{y} \in C$ をとってくる。
任意の $k \in \{1, 2, 3, \cdots, m\}$ に対して
$x_k = y_k \Longrightarrow x_k = y_k = z_k$
$x_k \neq y_k \Longrightarrow x_k = z_k, y_k \neq z_k \ or \ x_k \neq z_k, y_k = z_k$
となるように, かつ $x_k \neq y_k$のとき $x_k \neq z_k$となる回数が e となるように

$\mathbf{z} \in (\mathbf{F}_q)^m$を定める。作り方からそのようなものは必ず存在する。このとき
$\mathbf{z} \in B(\mathbf{x}, e) \quad d(\mathbf{x}, \mathbf{z}) = e \quad d(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$
となり, $\mathbf{z} \in B(\mathbf{x}, e)$ であることから $\mathbf{z} \notin B(\mathbf{y}, e)$ なので $d(\mathbf{z}, \mathbf{y}) \geq e + 1$
従って
$d(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y}) \geq 2e + 1$
よって $min\ dist\ of\ C = min\{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \neq \mathbf{y}, \mathbf{x}, \mathbf{y} \in C\} \geq 2e + 1$
次に $\Longleftarrow$ を示す。
$B(\mathbf{x}, e) \cap B(\mathbf{y}, e) \neq \phi$とする。
このとき, ある $\mathbf{z} \in B(\mathbf{x}, e) \cap B(\mathbf{y}, e)$ が存在して
$d(\mathbf{x}, \mathbf{z}) \leq e \quad$ かつ $\quad d(\mathbf{z}, \mathbf{y}) \leq e$
となり, 三角不等式により
$d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y}) \leq 2e$
となり矛盾する。

距離関数の性質 距離関数 d の定義から次のことが明らかに成り立つ

$$d(\mathbf{x}, \mathbf{y}) = d(\mathbf{x} - \mathbf{y}, \mathbf{0})$$

$C \subset (\mathbf{F}_q)^m$ を $\mathbf{F}_q$-vector space とすると
$$\begin{aligned} min\ dist\ of\ C &= \{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \neq \mathbf{y},\ \mathbf{x}, \mathbf{y} \in C\} \\ &= \{d(\mathbf{x}, \mathbf{0}) \mid \mathbf{x} \neq \mathbf{0},\ \mathbf{x} \in C\} =: min\ wt\ of\ C \end{aligned}$$
となる。

以下 $C \subset (\mathbf{F}_q)^m$を $\mathbf{F}_q - vector\ \ space$ とする。

定理 1.7. $C \subset (\mathbf{F}_q)^m,\ dimC = d$ とする。このとき $(m - d) \times m$ 行列で KerP=C となる行列 P がいくつも存在する。

証明 C の基底を $\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3}, \cdots, \mathbf{b_d}$ とする。
$$\begin{pmatrix} {}^{\mathrm{t}}\mathbf{b_1} \\ {}^{\mathrm{t}}\mathbf{b_2} \\ \vdots \\ {}^{\mathrm{t}}\mathbf{b_d} \end{pmatrix} \mathbf{x} = \mathbf{0}$$ の解空間は次元公式により $m - d$ 次元である。
解空間の基底を $\mathbf{p_1}, \mathbf{p_2}, \cdots, \mathbf{p_{m-d}}$ととると
$$\begin{pmatrix} {}^{\mathrm{t}}\mathbf{b_1} \\ {}^{\mathrm{t}}\mathbf{b_2} \\ \vdots \\ {}^{\mathrm{t}}\mathbf{b_d} \end{pmatrix} \begin{pmatrix} \mathbf{p_1} & \mathbf{p_2} & \cdots & \mathbf{p_{m-d}} \end{pmatrix} = \mathbf{0}$$

$\begin{pmatrix} \mathbf{p_1} & \mathbf{p_2} & \cdots & \mathbf{p_{m-d}} \end{pmatrix} = {}^t P$ とおく。
すると
$P \begin{pmatrix} \mathbf{b_1} & \mathbf{b_2} & \cdots & \mathbf{b_d} \end{pmatrix} = \mathbf{0}$
C の基底が P の Ker に入っているので
$C \subset KerP$
ここで再び次元公式により $dimKerP = d$, したがって $dimC = dimKerP$
よって $C = KerP$ となる。このような P が複数存在することは解空間の基底の取り方により P が変わることから明らかであろう。

定義 1.8. 定理 1.7. の行列 P を C の Parity check 行列という。

定理 1.9. $min\ wt\ of\ C = \{P$ の列で 1 次従属となる組の最小個数 $\}$

証明 $\delta = min\ wt\ of\ C$ とおく。
$P = (\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_m}) \qquad \mathbf{x_i} \in (\mathbf{F}_q)^{m-d}$ とおき,
$\mathbf{y} \in C = KerP$ がδを与えたとする。つまり $d(\mathbf{y}, \mathbf{0}) = \delta$, これは
$\mathbf{y} = (y_1, y_2, \cdots, y_m)$ の成分のうちδ個を除いて 0 ということである。
$\mathbf{y} \in KerP$ より

$$(\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_m}) \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = y_1\mathbf{x_1} + y_2\mathbf{x_2} + \cdots + y_m\mathbf{x_m} = \mathbf{0}$$

よって $\mathbf{x_1}$から $\mathbf{x_m}$のうちδ個を選んで 1 次従属にできる。
逆に $\mathbf{x_1}$から $\mathbf{x_m}$のうち k 個選んで 1 次従属にできたら
ある $\mathbf{z} \in KerP$ が存在して $d(\mathbf{z}, \mathbf{0}) = k \geq \delta$

系 1.10 P のどの $min\ wt\ of\ C - 1$ 個以下の列も 1 次独立である。

証明 定理 1.9. より明らか

2 error correcting sequence

定義 2.1. S を $\mathbf{F}_q$ 上の数列で周期が p であるものとする。

$$Sh_m(S) := \{(x_i, x_i + 1, \cdots, x_{i+m-1}) \mid i = 0, 1, \cdots, p-1\} \subset (\mathbf{F}_q)^m \ (m \in \mathbf{N})$$

が符号長 m の e-error correcting code となるとき S を符号長 m の
e-error correcting sequence であるという。

定義 2.2. $\mathbf{F}_q$ 上の数列 S が n 次のM 系列であるとは S が n 階漸化式

$$x_{n+i} + a_{n-1}x_{n+i-1} + \cdots + a_0 x_i = 0 \ (a_j \in \mathbf{F}_q) \cdots (2.2)$$

の解であって周期が $q^n - 1$ となるもの

定義 2.3. $\varphi(t) \in \mathbf{F}_q[t], deg\varphi(t) = k$ が原始多項式であるとは,$\varphi(t)$ が $\mathbf{F}_q$ 上既約で, 乗法群 $(\mathbf{F}_q[t]/\varphi(t))^\times$ における t の乗法位数が $q^k - 1$, すなわち乗法群を生成すること。

定理 2.4.

$$x_{n+i} + a_{n-1}x_{n+i-1} + \cdots + a_0 x_i = 0 \ (a_j \in \mathbf{F}_q)$$

で定義された $\mathbf{F}_q$ 上の数列がM系列となるための必要十分条件は

$$\varphi(t) = t^n + a_{n-1}t^{n-1} + \cdots + a_0$$

が原始多項式になることである。

定理 2.5. S を $\mathbf{F}_q$上の n 次の M 系列とする。$Sh_m(S) \cup \{\mathbf{0}\}$ は
$\mathbf{F}_q - vector\ space$ である。

証明 S が (2.2) で定義されているとする。n 次の M 系列であることから $m \leq n$ のときは明らか。$m > n$ とする。$\mathbf{x} = (x_m, x_{m-1}, \cdots, x_1)$ とするとき
$\mathbf{x} \in Sh_m(S) \cup \{\mathbf{0}\}$
$\iff x_{n+i} + a_{n-1}x_{n+i-1} + \cdots + a_0 x_i = 0 \qquad \forall i \in \{1, 2, \cdots, m-n\}$
である。
任意の $\mathbf{x}, \mathbf{y} \in Sh_m(S) \cup \{\mathbf{0}\}$ に対し
$\mathbf{x} = (x_m, x_{m-1}, \cdots, x_1)$
$\mathbf{y} = (y_m, y_{m-1}, \cdots, y_1)$
とおく。さっき見たことから任意の $i \in \{1, 2, \cdots, m-n\}$ に対し
$x_{n+i} + a_{n-1}x_{n+i-1} + \cdots + a_0 x_i = 0$
$y_{n+i} + a_{n-1}y_{n+i-1} + \cdots + a_0 y_i = 0$
この二つの式を辺々たすと
$(x_{n+i} + y_{n+i}) + a_{n-1}(x_{n+i-1} + y_{n+i-1}) + \cdots + a_0(x_i + y_i) = 0$
となるので,$\mathbf{x} + \mathbf{y} \in Sh_m(S) \cup \{\mathbf{0}\}$ また任意の $k \in \mathbf{F}_q$に対して
$x_{n+i} + a_{n-1}x_{n+i-1} + \cdots + a_0 x_i = 0$
$\iff (kx_{n+i}) + a_{n-1}(kx_{n+i-1}) + \cdots + a_0(kx_i) = 0$
なので $k\mathbf{x} \in Shm(S) \cup \{\mathbf{0}\}$ である。従って $Sh_m(S) \cup \{\mathbf{0}\}$ は
$\mathbf{F}_q - vector\ space$ である。

補題 2.6. $m > n$ とする。数列 S を (2.2) で定められたM系列とする。このとき

$$P := \underbrace{\left.\begin{pmatrix} a_0 & a_1 & \cdots & a_{n-1} & 1 & 0 & \cdots & \cdots & 0 \\ 0 & a_0 & a_1 & \cdots & a_{n-1} & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & & 0 & a_0 & a_1 & & & 1 & 0 \\ 0 & \cdots & \cdots & 0 & a_0 & a_1 & \cdots & a_{n-1} & 1 \end{pmatrix}\right\} m-n}_{m}$$

は $Sh_m(S) \cup \{\mathbf{0}\}$ の Parity check 行列となる。

証明 $KerP = Sh_m(S) \cup \{\mathbf{0}\}$ を示せばよい。$\supset$ は明らか。
$\subset$ を示す。
S が n 次の M 系列であるから$\#Sh_n(S) = q^n - 1$ である。このことと
$m > n$ から$\#Sh_m(S) = q^n - 1$ であることが分かる。
$rankP = m - n$ であるから次元公式により $dimKerP = n$ である。従って
$\#KerP = q^n = \#Sh_m(S) \cup \{\mathbf{0}\}$ である。$Sh_m(S) \cup \{\mathbf{0}\} \subset KerP$ で元の個数が等しいので $Sh_m(S) \cup \{\mathbf{0}\} = KerP$ である。

命題 2.7. $Sh_m(S)$ が $e-error\ correcting\ code$ となる必要十分条件は P のどの $2e$ 列も 1 次独立となることである。

証明 $Sh_m(S)$ が $e - error\ correcting\ code$
$\iff min\ wt\ of\ Sh_m(S) \geq 2e+1$(補題 1.6 より)
$\iff min\ wt\ of\ Sh_m(S) \cup \{\mathbf{0}\} \geq 2e+1$
$\iff P$ のどの $2e$ 列の 1 次独立である。(系 1.10 より)

例 $min\ wt\ of\ Sh_m(S) = \delta$とする。
$\delta \geq 1 \cdots$自明
$\delta \geq 2 \iff P$ のどの 1 個の列も 1 次独立 $\iff P$ に 0 からなる列がない。
$\delta \geq 3 \iff P$ に 0 からなる列がなく, かつ P のどの 2 列も 1 次独立
$Sh_m(S) \subset (\mathbf{F}_2)^m$　の場合
$\delta \geq 3 \iff P$ に 0 からなる列がなく, かつ P のどの列も異なる。
$\iff$ 周期 m の数列 L
$\overbrace{\underbrace{00\cdots0a_0}_{m-n} a_1 \cdots a_{n-1} 1}^{m} 0 \cdots$
において $Sh_{m-n}(L)$ の元が全て異なり, $\mathbf{0}$ がない。

以上のことから $q=2$ のときの $1-error\ correcting\ sequence$ の生成法を示す。

周期 m の数列 L

$\overbrace{\underbrace{00\cdots0a_0}_{m-n}a_1\cdots a_{n-1}1}^{m}0\cdots$

において $Sh_{m-n}(L)$ の元が全て異なり,

0 がないような $a_j(j=1,2,\cdots n-1)$ を探す。

で探した a_j に付随する多項式

$t^n+a_{n-1}t^{n-1}+\cdots+a_0$ が $\mathbf{F}_2$ 上の原始多項式であることをしらべる。

, が満たされたならば n 階漸化式

$x_{n+i}+a_{n-1}x_{n+i-1}+\cdots+a_0x_i=0$ により周期 2^n-1 で符号長 m の

$1-error\ correcting\ sequence$ が生成される。

m を固定したときに周期最大, つまり n が最大の $1-error\ correcting\ sequence$ を生成することを考察する。

$m=2^k$ のとき $n=2^k-k$ として $m-n=k$ 次の $DeBruijn$ 系列 L をつくると $Sh_{m-n}(L)$ の元が全て異なる。これ以上 n が大きくなると $Sh_{m-n}(L)$ の元で重複するものができる。なぜなら $n>2^k-k$ なら $m-n<k$ となり $2^{m-n}<2^k=m$ となるから。

しかし $m-n=k$ 次の $DeBruijn$ 系列の $Sh_{m-n}(L)$ には $\underbrace{00\cdots0}_{m-n}$ が含まれている。そして $DeBruijn$ 系列に 1 は $2^{m-n}/2=2^{m-n-1}$ 個含まれているのでこれに付随する多項式は 1 を根にもつ。従って原始的でない。この 2 つのことから $DeBruijn$ 系列から $0,1$ を 1 個ずつ除きたい。除いたときに $DeBruijn$ グラフが一筆描きが出来るようにするには $\underbrace{00\cdots0}_{m-n-1}$ をシフトして 0 を加えたときに出る辺と, $\underbrace{11\cdots1}_{m-n-1}$ をシフトして 1 を加えたときに出る辺を除くしかない。$DeBruijn$ グラフからこの 2 個の辺を除いたときのグラフを $sub-DeBruijn$ グラフということにし, これを一筆描きしたときに出来る系列を $sub-DeBruijn$ 系列ということにする。従って $m=2^k-2$ とすると n の最大として $n=2^k-k-2$ がとれて $m-n=k$ 次の $sub-DeBruijn$ 系列を見つけ出し, それに付随する多項式が原始多項式になることをチェックすればよい。

以上をまとめると k 次の $sub-DeBruijn$ 系列に付随する多項式が原始多項式になれば周期 $2^{2^k-k-2}-1$ で符号長 $m=2^k-2$ の

$1-error\ correcting\ sequence$ が作れる。

$q=2$ のとき k 次の $sub-DeBruijn$ 系列の個数とそれに付随する多項式のうち原始多項式となるものの個数を示す。

k	$sub-DeBruijn$ の個数	原始多項式の個数
2	1	0
3	2	2
4	16	2
5	2048	316

C 言語による N 次の $sub-DeBruijn$ 系列の生成, またそれに付随する多項式が原始多項式かを調べるソースプログラムを示す。原始性の判定は定義通りの求め方であるから時間のかかる悪いプログラムであると思う。

```
#include<stdio.h>
#define true (0)
#define false (1)
#define N (5)
#define Q (2)
#define P (10000)
#define M (100)

unsigned long int power(int x,int y)
{
        int i;
        int mul=1;
        for(i=1;i<=y;i++)
               mul*=x;
        return mul;
}
int pre(const int c[])/*原始性を判定する関数*/
{                       /*受け渡しの配列は各項の係数である*/
        unsigned long int deg=2;
        unsigned long int de=1;
        unsigned long int i;
        unsigned long int n=power(Q,N)-N-2;/*次数*/
        for(i=2;i<=n+1;i++)
        {
                de=de<<1;
                de=de+c[i];
```

```
        }
        for(i=1;i<=power(Q,n)-2;i++)
        {
                deg=deg<<1;
        if(deg>=power(Q,n))
                {
                        deg=deg^de;
                        if(deg==1)
                        {
                               if(i==power(Q,n)-2)
                               {
                                      return 1;
}
                                else
                                {
                                      return 0;
                                }
                         }
                 }
       }
         return 0;
}

int count=0;/*sub DeBruijn系列の個数をカウントする変数*/
int precount=0;/*原始的なものの個数をカウントする変数*/
void deb(int dx)/*sub DeBruijn系列を生成する関数*/
{
        static unsigned long int j=1;
        int i;
        static int d[P]={false,true};
        static int pos[P]={0,1};
        d[power(Q,N)-1]=false;

        if(dx==0 && j==power(Q,N)-1)
        {
```

```
                precount=precount+pre(pos);
                j--;
                count++;
                return;
        }

        if(d[dx]==true)
        {
                d[dx]=false;
                for(i=0;i<Q;i++)
                {
                        pos[++j]=i;
                        deb((dx*Q+i)%power(Q,N));
                }
                d[dx]=true;
         }
         j--;
}
int main(void)
{
deb(1);
printf("\n");
printf("%d , %d",count,precount);
printf("\n");
return(0);
}
```