18 ❦ 学籍番号を QR コードにしよう (2)

2023年度　総合教育科目演習Ｉ（木２限）

入学年度		学部	学 科		組		番 号			検	フリガナ	
2	3	B		1							氏 名	

● 誤り訂正符号化

前回，学籍番号から 13byte（8bit の組 13 個）からなる情報語を作った．今回はこれに誤り訂正コード語を加えて符号語をつくるところから始める．

1-Q 型の QR コードでは RS 符号と呼ばれる符号を用いる．RS 符号は $GF(2^8) = GF(256) = \mathbf{F}_{256}$ という数の体系を用いて作られる．$\mathbf{F}_{256}$ は $GF(2) = \mathbf{F}_2$ に

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

をみたす γ という “虚数” を付け加えた数の体系である．$\mathbf{F}_{256}$ の数は γ の 7 次以下の多項式で表され，8 bit = 1 byte の情報を保持する．また，$\mathbf{F}_{256}$ の 0 以外の数は γ^k $(k = 0, 1, \ldots, 254)$ と表せることに注意しておく（乗法表示）．

BCH 符号では，情報を 0 と 1 を係数に持つ多項式，すなわち “1 bit” 係数の多項式で表し，それに剰余などの代数的操作を加えて符号語を作るのであった．これに対し，RS 符号では，係数が “1 byte” である多項式に同様の操作を用いて誤り訂正符号を作る．

ここで用いる RS(26, 13) は $\mathbf{F}_{256}$ を係数とする 25 次多項式を符号語とする符号である．前回作ったデータは 13 byte あるが，その各 byte を γ の 7 次以下の多項式とみなし，$\mathbf{F}_{256}$ の数とみなす．たとえば，1 行目の “00100000” は γ^5，2 行目の “01011000” は $\gamma^6 + \gamma^4 + \gamma^3$ などとする．そして，この 13 byte の情報語を，係数が $GF(2^8)$ の要素である x の 13 次の多項式とみなす．すなわち，上の情報語は

$$q(x) = \gamma^5 x^{13} + (\gamma^6 + \gamma^4 + \gamma^3)x^{12} + \cdots + (\gamma^7 + \gamma^6 + \gamma^5 + \gamma^3 + \gamma^2)$$

という情報多項式で表せる．

BCH 符号では，情報多項式 $q(x)$ から生成多項式 $g(x)$ を用いて送信多項式を作るのであった．RS 符号でも，BCH 符号と同じ要領で送信多項式を作る．RS(26, 13) では，生成多項式 $g(x)$ を

$$g(x) = (x + 1)(x + \gamma)(x + \gamma^2)(x + \gamma^3) \times \cdots \times (x + \gamma^{12})$$

として，送信多項式 $u(x)$ を $g(x)$ 用いて次のようにする．

$$u(x) = q(x)x^{13} + \bigl(q(x)x^{13} \text{ を } g(x) \text{ で割った余り}\bigr)$$

$u(x)$ を計算するために，Mathematica で次のようなファイルを作って計算する．ただし，途中の “□” には各自のデータを入力すること．

こうして得られた送信語を裏の表に写す．

```
生成元 = γ^8 + γ^4 + γ^3 + γ^2 + 1;
加法表示[x_] := PolynomialMod[x, 生成元, Modulus -> 2]
F256 = Prepend[Table[加法表示[γ^k], {k, 0, 254}], 0];
位置[L_, e_] := Position[L, e][[1]][[1]];
乗法表示[x_] := If[x === 0, 0, γ^(位置[F256, 加法表示[x]] - 2)];
生成多項式 = PolynomialMod[Product[(x + γ^k), {k, 0, 12}], 生成元,
 Modulus -> 2];

情報多項式 = Map[加法表示[#.Table[γ^(7 - i), {i, 0, 7}]] &,
    {{0, 0, 1, 0, 0, 0, 0, 0},
     {0, 1, 0, 1, 1, 0, 0, 0},
     {□, □, □, □, □, □, □, □},
     {□, □, □, □, □, □, □, □},
     {□, □, □, □, □, □, □, □},
     {□, □, □, □, □, □, □, □},
     {□, □, □, □, □, □, □, □},
     {□, □, □, □, □, □, □, □},
     {□, □, □, □, □, □, □, □},
     {□, □, □, □, □, □, □, □},
     {□, □, □, □, □, □, □, □},
     {□, □, □, □, □, □, □, □},
     {1, 1, 1, 0, 1, 1, 0, 0}}
    ].Table[x^(13 - i), {i, 1, 13}];

符号多項式 =
  Collect[PolynomialMod[
    情報多項式*x^13 + PolynomialRemainder[情報多項式*x^13, 生成多項式, x],
    生成元, Modulus -> 2], x];

送信語 = Mod[
   Map[Table[Coefficient[加法表示[#], γ, 7 - i], {i, 0, 7}] &,
    Table[Coefficient[符号多項式, x, 25 - i],
     {i, 0, 25}]], 2];
ExportString[Table[{i, 送信語[[i]]}, {i, 1, 26}], "Table"]
```

1.	0	0	1	0	0	0	0	0
2.	0	1	0	1	1	0	0	0
3.								
4.								
5.								
6.								
7.								
8.								
9.								
10.								
11.								
12.								
13.	1	1	1	0	1	1	0	0
14.								
15.								
16.								
17.								
18.								
19.								
20.								
21.								
22.								
23.								
24.								
25.								
26.								

● マスク処理

マスク処理のために，マスクパターンに対応した 8 bit データを送信語の各語に加えていく．もちろん「加える」ときには，$1+1=0$ として計算する．このような演算は「排他的論理和」と呼ばれることがある．ここでは，正式な仕様には基づかず，予め一番簡単な市松模様のマスクを用いると決めてしまうことにする．少々面倒ではあるが，この計算を右上のページで手作業でやってみよう．

● 形式情報

ここまで作成した QR コードの誤り訂正レベルは「Q」= 11，マスクパターンは「000」であるので，これを記述する形式情報をつくる．これは BCH(15, 5) 符号を用いて行われる．まず，情報語 11000 から，情報多項式 $q(x)=x^4+x^3$ を作り，それに x^{10} をかけ，それ生成多項式 $g(x)=x^{10}+x^8+x^5+x^4+x^2+x+1$ で割った余りを加えるのであった．これを実行してみると，

$$u(x)=x^{14}+x^{13}+x^8+x^6+x^3+x^2+1$$

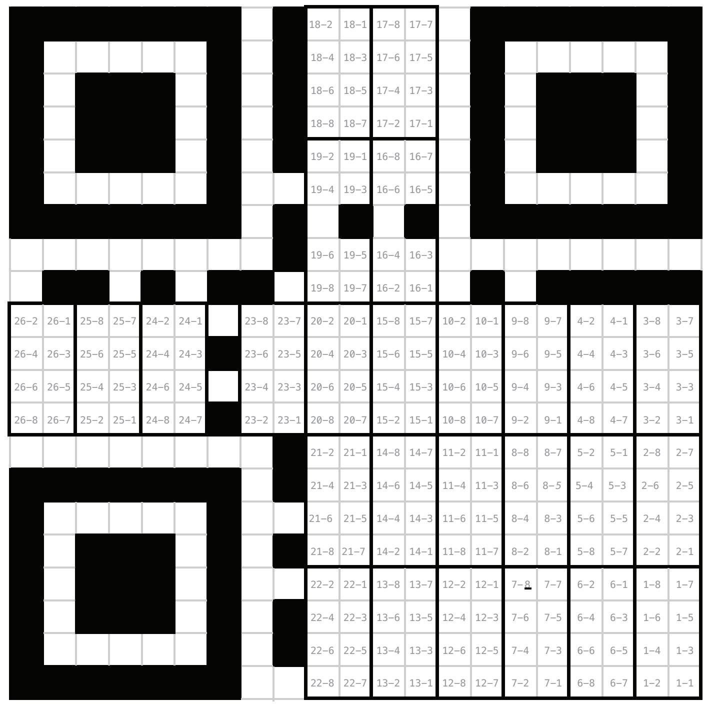
となる．これを符号語に直すと 110000101110011 となる．これに形式情報のマスク 101010000010010 との排他的論理和をとと，

$$\begin{array}{rr} & 110000101001101 \\ +) & 101010000010010 \\ \hline & 011010101011111 \end{array}$$

この結果は右下の図にすでに反映されている．

位置	送信語	マスク	排他的論理和
1		1 0 0 1 1 0 0 1	
2		1 0 0 1 1 0 0 1	
3		1 0 0 1 1 0 0 1	
4		0 1 1 0 0 1 1 0	
5		0 1 1 0 0 1 1 0	
6		0 1 1 0 0 1 1 0	
7		1 0 0 1 1 0 0 1	
8		1 0 0 1 1 0 0 1	
9		1 0 0 1 1 0 0 1	
10		0 1 1 0 0 1 1 0	
11		0 1 1 0 0 1 1 0	
12		0 1 1 0 0 1 1 0	
13		1 0 0 1 1 0 0 1	
14		1 0 0 1 1 0 0 1	
15		1 0 0 1 1 0 0 1	
16		1 0 0 1 1 0 0 1	
17		0 1 1 0 0 1 1 0	
18		1 0 0 1 1 0 0 1	
19		1 0 0 1 1 0 0 1	
20		0 1 1 0 0 1 1 0	
21		0 1 1 0 0 1 1 0	
22		0 1 1 0 0 1 1 0	
23		1 0 0 1 1 0 0 1	
24		1 0 0 1 1 0 0 1	
25		0 1 1 0 0 1 1 0	
26		1 0 0 1 1 0 0 1	

この結果をもとに右のページの上の図のマス目を黒く塗っていく．



担当：鍬田 政人 (kuwata.04f@g.chuo-u.ac.jp)