前々回，1 つの誤りを訂正可能なハミング符号 $H(7,4)$ を扱い，前回それは BCH 符号と呼ばれる誤り訂正の仕組みと同等であることを示した．

有限体 $\mathbb{F}_{2^q}$ を利用する BCH 符号では，(2^q-1) bit の情報を (2^q-2) 次多項式と捉え，そこに送りたいデータ（**情報語**）と誤り訂正用のデータ（**検査語**）を詰め込んだ**符号語**として送受信する．ハミング符号 $H(7,4)$（7 は符号語の bit 数，4 は情報語の bit 数を表す）は $\mathbb{F}_8$ を用いた BCH 符号 $(7,4)$ と同等になる．

ここでは $\mathbb{F}_8$ を用いた $(7,4)$ 型と呼ばれる最も簡単な BCH 符号について，具体例を用いて詳しく見ることにする．実は，この $\mathbb{F}_8$ を用いた BCH 符号は前回のハミング符号と全く同値なものになるのであるが，誤り訂正の仕組みには有限体 $\mathbb{F}_8$ の性質が用いらる．

$\mathbb{F}_8$ は $\mathbb{F}_2$ に $\alpha^3+\alpha+1=0$ をみたす数 α を加えて得られる数の体系であった．$\mathbb{F}_8$ の元はすべて α の 2 次以下の多項式として表され（**加法表示**），0 を除く 7 個の元は α^k（$0 \leqq k \leqq 6$）と表せる（**乗法表示**）のであった．まず，その対応を Mathematica™ でプログラミングする．（Mathematica™ における関数の定義の仕方はヘルプを参照のこと．引数 poly は α の多項式である必要がある．“additive” とか，“multiplicative” という関数名は好みにより変更できる．）

```
additive[poly_] := PolynomialMod[poly, α^3 + α + 1, Modulus -> 2];

F8 = Prepend[Table[additive[α^k], {k, 0, 6}], 0];
     (* 𝔽8の元のリストを作る．最初に 0もふくめてある．*)

multiplicative[poly_] := If[additive[poly] === 0, 0,(* 0は 0を返す．*)
  α^(Position[F8, additive[poly]][[1]][[1]] - 2)];
    (* 𝔽8の表のどこにあるかを逆引き．Position の仕様により指数を調整してある．*)
```

BCH 符号では符号語を x の多項式として扱う（符号多項式と呼ぶことにする）．BCH(7, 4) では，長さ 7 の符号語を 6 次多項式と見做す．BCH 符号の基本的考え方は，長さ 7 の語が符号語であるためには，対応する 6 次の符号多項式 $u(x)$ が $g(x)=x^3+x+1$ で割り切れると定義することである．これは，$f(x)$ が $f(\alpha)=0$ を満たすことと同値である．

● 符号化

まず，送信したい 4bit の情報語を 3 次以下の情報多項式 $q(x)$ に変換する．符号多項式 $u(x)$ を作るには，$u(x)$ に x^3 を掛けて 6 次式とし（桁をシフトすることに相当する），それが生成多項式 $g(x)$ で割りきれるように余りを加える．すなわち，次のように定義する．

情報多項式：　$q(x)=$ (3 次以下の多項式),
生成多項式：　$g(x)=x^3+x+1$,
符号多項式：　$u(x)=q(x)x^3+\big(q(x)x^3$ を $g(x)$ で割った余り $\big)$

この定義により，“正しい” 多項式である符号多項式 $u(x)$ は $g(x)$ で割り切れる多項式ということになり，$u(x)=g(x)q_0(x)$ の形にかけ，$x=\alpha$ を代入すると，$u(\alpha)=0$ となる．

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

これを Mathematica™ で計算するには次のようにすればよい．ここで，引数 “info” には “0101” のように 4 桁の数を文字列として入力する．（最初の桁が 0 になるときをケアするため．）

```
u[info_] := Module[{qx},
  qx = ToExpression[Characters[info]].Table[x^k, {k, 3, 0, -1}];
  Expand[qx*x^3 + PolynomialMod[qx*x^3, x^3 + x + 1, Modulus -> 2], Modulus -> 2]]
```

ここまでで，次のような結果が得られるはずである．（Input, Output の番号はその都度変わる．）

```
In[5]:= u["1101"] ⇧+↵

Out[5]= 1 + x^3 + x^5 + x^6

In[6]:= additive[Out[45] /. {x -> α}] ⇧+↵

Out[6]= 0
```

● 復号（誤り訂正）

符号語が通信経路を通って受信語として受信されたとき，まずそれを受信多項式 $r(x)$ に変換する．通信経路で誤りが起こったとして，その誤差を $e(x)$ とおくと

$$r(x)=u(x)+e(x)$$

と書ける．この $e(x)$ は**誤差多項式**と呼ばれる．ここで，符号語と受信語は高々 1bit しか相違していないと仮定する．すると，

$$e(x)=0 \quad \text{または} \quad e(x)=x^k \quad (0 \leq k \leq 6)$$

という形をしているはずである．いま，符号多項式 $u(x)$ に α を代入すると $u(\alpha)=0$ となるのであったから，受信多項式 $r(x)$ に α を代入すると

$$r(\alpha)=u(\alpha)+e(\alpha)=0+e(\alpha)=e(\alpha)$$

が成り立つ．ここで，$r(\alpha)=e(\alpha)=s$ とおき，s を「シンドローム（症候）」と呼ぶ．

シンドローム $s=e(\alpha)$ は，0 または α^k に等しいので，$r(\alpha)$ を計算してそれを乗法表示して α^k の形に直しておく．そして，

- $s=0$ なら誤りなし，
- $s=\alpha^k$ であれば，$r(x)$ と $u(x)$ は k 次の項が異なることを示す．すなわち，受信語の右から $k+1$ 番目の bit に誤りがる．

例えば，1011011 という語を受信したとすると

$$r(\alpha)=\alpha^6+\alpha^4+\alpha^3+\alpha+1=\alpha+1=\alpha^3$$

だから，x^3 の係数の 1 が誤りでこれが 0 でなければならず，正しい符号語は 1010011 とわかる．

【裏に続く】

```
correction[received_] := Module[{ra, ua},
  ra = ToExpression[Characters[received]].Table[α^k, {k, 6, 0, -1}];
  ua = Expand[ra + multiplicative[ra], Modulus -> 2];
  IntegerString[FromDigits[Reverse[CoefficientList[ua, α]]], 10, 7]]
```

1 (7, 4) 型の BCH 符号で 0101001 という語を受信したとする．誤りは高々 1 個であるという仮定の下に情報語を求めよ.

BCH(15, 11)

次に，$\mathbb{F}_8$ の代わりに $\mathbb{F}_{16} = GF(2^4)$ を用いて BCH 符号を構成することを考える.

$\mathbb{F}_{16}$ は $\mathbb{F}_2$ の $\beta^4 + \beta + 1$ を満たす元 β を付け加えた体であり，$\mathbb{F}_{16}$ の元はすべて β の 3 次以下の多項式として表される（加法表示）．また，$\mathbb{F}_{16}^{\times}$ の元は β^k（$0 \leqq k \leqq 14$）と表せる（乗法表示）.

BCH(7, 4) にならって，符号語は生成多項式 $g(x) = x^4 + x + 1$ で割り切れるものと定義すると，検査 bit は生成多項式の余りの次数 +1 なので，4bit となる．したがって，$\mathbb{F}_{16}$ を用いて 1 つの誤りが訂正可能な BCH 符号は符号語の bit 数が 15，11 は情報語の bit 数は $15 - 4 = 11$ である (15, 11) 型となる．すなわち，情報語は 10 次多項式で表され，それを 14 次式の符号多項式 $u(x)$ に変換して送受信する.

2 BCH(7, 4) の Mathematica™ のコードを流用して，BCH(15, 11) のものに直し，11bit の情報から 15bit の符号語を作ったり，15bit の語を訂正して正しい符号語に訂正できるようにせよ.

担当：鍬田 政人 (kuwata.04f@g.chuo-u.ac.jp)