組合せ理論の応用への入門

組合せ理論というのはきわめて広い分野で，その応用の全貌をとらえることは筆者は任にたえないので，ここではその中でガロア体というものの応用だけを取りあげてみることにしたい．事実組合せ理論の中のとくに実用的な応用の面ではガロア体の活躍はめざましく，ガロア体だけに限っても話題は結構豊富になるのである．したがって表題はむしろ「ガロア体の応用」というほうが内容にふさわしいかも知れないが，なじみのない読者も多いかと思って表記のものとした．定理などの証明は省略したが例題を通して納得しうると思う．

1. ガロア体とは

ガロア体というのは一口でいえば，実数の四則と同一特性の算法をもつ有限集合であるといえる．その意味で有限体とよぶ人もある．元の数が s 個の，つまり大きさ s の，ガロア体を $GF(s)$ と書くが，s がちょうど素数 p のときつまり $GF(p)$ はもっとも簡単なガロア体で，mod p の算法と同一のものである．つまり $\{0, 1, \cdots, p-1\}$ の中で普通の整数としての加法乗法を行ない結果が p 以上になれば p で割った余りをとるという算法を考えればこれが $GF(p)$ をつくる．

たとえば $GF(5)$ の加法乗法は**表1**のようになる．$a+x=0$ なる x を $-a$(a の負数)，$ax=1$ なる x を a^{-1}(a の逆数)と考え，これによって減法，除法が定義できる．実数の四則の特性の中で，0以外の各数がすべて逆数をただ1つもつ，という性質が $GF(5)$ にもあることは表1から容易にわかる．もし s が素数でないと mod s の算法を考えてもこの性質が成り立たなくなるのである（mod 6 をためしてみられるとよい）．

s が素数でなくても素数の累乗つまり $s=p^r$ なら，[3][5]などにあるように上の方法をやや拡張した方法で，大きさ s のガロア体 $GF(s)=GF(p^r)$ がつくれる．しかしそれ以外の自然数 s に対して大きさ s の有限体は残念ながら存在しないのである．

ここでは簡単のため $GF(p)$ に限ってその応用の話をしよう．

表 1 GF(5)の加法，乗法

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

2. 直交実験の構成

たとえばある化学製品の歩留り y に影響を与える因子として，反応温度(A)，炉(B)，触媒(C)が考えられ，各因子の水準が，

$$A\begin{cases}0:800^\circ\\1:900^\circ\end{cases}\quad B\begin{cases}0:第1炉\\1:第2炉\end{cases}\quad C\begin{cases}0:触媒無\\1:触媒有\end{cases}\qquad(1)$$

であるとする．

一般にある特性値 y に影響を与える因子が上のように離散的な水準で特徴づけられている場合の実験を要因配置実験[1]とよんでいる[1]．われわれの目的は各因子の水準が特性値にどのような影

響を与えているかを実験によって知ることにある．

このときすべての水準組合せを r 回* ずつ行なう実験を完全実験とよぼう（*因子数が少ないときは $r\geqq 2$ のときもあるが通常 $r=1$）．完全実験を行なえば当然完全な情報が得られると考えられるが，因子数が10各因子がすべて3水準といった程度の場合でも，実験回数は $3^{10}\fallingdotseq 6$ 万回となる．これではたまらないからといって多くの技術者は，あまり重要でないと思われる因子の水準を1つに固定して実験回数を減らそうとするが，そうするときわめて偏った情報しか得られず，対象に対して誤った判断を下す恐れがある．

表 2

	A	B	C
1	0	0	0
2	1	0	0
3	0	1	0
4	0	0	1
5	1	1	0
6	0	1	1
7	1	0	1
8	1	1	1

もっと少ない実験で偏らない情報が得られないだろうかという考えは自然な要求である．いま**表2**の完全実験から4つの水準組合せを選ぶ3通りの方法を**表3**に示した．表3(イ)は上の技術者のように因子 C の水準を0に固定したやり方で，(ロ)(ハ)はさらに工夫した選び方である．いま各実験の中で各水準の出現頻度を数えてみよう．

$$\left.\begin{array}{ll}(イ) & A_0=2\quad B_0=2\quad C_0=4\\ & A_1=2\quad B_1=2\quad C_1=0\\ (ロ) & A_0=2\quad B_0=2\quad C_0=2\\ & A_1=2\quad B_1=2\quad C_1=2\end{array}\right\}\qquad (2)$$

（A の0，1水準の出現頻度をそれぞれ A_0, A_1 としてある B_i，C_i も同様），(ハ)は(ロ)と同じである．

表 3

(イ)

	A	B	C
1	0	0	0
2	1	0	0
3	0	1	0
5	1	1	0

(ロ)

	A	B	C
1	0	0	0
3	0	1	0
7	1	0	1
8	1	1	1

(ハ)

	A	B	C
1	0	0	0
5	1	1	0
6	0	1	1
7	1	0	1

1) これに対して因子が計量的情報で特徴づけられるのが回帰分析とよばれている．上の例で温度は計量的ではあるが，ここでは800°，900°という量に固定して離散化して考えている．

このような頻度を1次の頻度とよび，これらが各因子の中で水準によらず一定である実験を，強さ1の直交実験とよぶことにしよう．(ロ)，(ハ)は強さ1の直交実験であるが，(イ)はそうでない．

さらに今度は2つの因子の水準組合せの出現頻度つまり2次の頻度を調べてみよう．

$$\left.\begin{array}{ll}(ロ) & (AB)_{00}=1\quad (AC)_{00}=2\quad (BC)_{00}=1\\ & (AB)_{01}=1\quad (AC)_{01}=0\quad (BC)_{01}=1\\ & (AB)_{10}=1\quad (AC)_{10}=0\quad (BC)_{10}=1\\ & (AB)_{11}=1\quad (AC)_{11}=2\quad (BC)_{11}=1\\ (ハ) & (AB)_{00}=1\quad (AC)_{00}=1\quad (BC)_{00}=1\\ & (AB)_{01}=1\quad (AC)_{01}=1\quad (BC)_{01}=1\\ & (AB)_{10}=1\quad (AC)_{10}=1\quad (BC)_{10}=1\\ & (AB)_{11}=1\quad (AC)_{11}=1\quad (BC)_{11}=1\end{array}\right\}\qquad (3)$$

（$(AB)_{00}$ は A が0で B が0である水準組合せの出現頻度，他の $(AC)_{ij}$ なども同様）．((イ)についても調べてみられたい）．(ロ)では $(AC)_{ij}$ が水準組合せ (ij) によって異なるが，(ハ)ではすべての因子対の中で2次の頻度が同一である．このような実験を強さ2の直交実験とよぶ．

すぐわかるように強さ2であれば必然的に強さ1でもある．(ハ)は強さ2，(ロ)は強さ1，(イ)は強さ1ですらない．このように見てくると表3で(ハ)(ロ)(イ)の順に良い実験であるといえよう．一般に同一実験回数であっても強さが強いほど良い実験であるといってよい．良いか悪いかの判定は，特性値の構造式によるが，構造式の中に交互作用効果[1]が存在しない場合，（実験回数が同一であれば）強さ2の直交実験が主効果[1]の推定に関して最良である（推定量の分散が一様に小さい）ことが証明されている[2]．

さて強さ2の直交実験をつくることは表2のように小さな例ならなんとかなるが，因子の数が多い場合には大変な仕事になる．それというのも(3)のような条件は組合せ論的なものであり一般にこの種のものは取扱いにくいからである．ところがガロア体を用いるとこれがきわめて簡単に解決する．

いま因子が m 個あって各因子とも水準が s であ

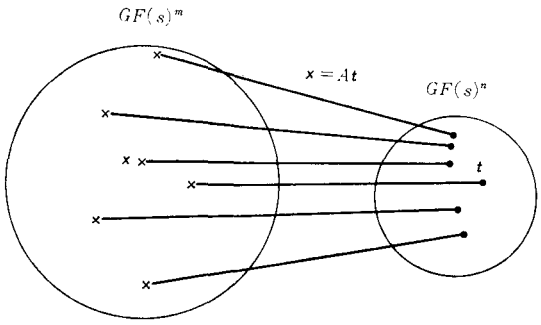


図 1

るとする.(水準数が因子によって異なる場合は以下のようにすっきりいかないが実用的な処理法はいろいろ考えられている). このとき実験回数 s^n の強さ2の直交実験をつくるには, $GF(s)$ 上[2] で

$$\left.\begin{aligned} &\boldsymbol{x}=[x_1,\cdots,x_m],\ \boldsymbol{t}=[t_1,\cdots,t_n]\\ &\boldsymbol{x}=\boldsymbol{t}A,\\ &A=\begin{bmatrix} a_{11}\cdots a_{1m}\\ \vdots\quad\ \vdots\\ a_{n1}\cdots a_{nm}\end{bmatrix},\ x_i, a_{ij}, t_j\in GF(s)\end{aligned}\right\} \quad (4)$$

としてつぎの定理によればよい.

定理1 A のどの2つの列ベクトルも $GF(s)$ 上1次独立であれば,

$$\Gamma=\{\boldsymbol{x}=\boldsymbol{t}A\,;\,\boldsymbol{t}\in GF(s)^n\} \quad (5)$$

が強さ2の直交実験を構成する. □

例として $m=4$ 因子, $s=3$ 水準の場合を考えよう. GF(3)は第1節の方法で計算できる. ガロア体は実数と四則性が同じだから, 1次独立といった線形代数的概念は実数の場合と同様に通用する.

$$A=\begin{bmatrix}1&0&1&1\\0&1&1&2\end{bmatrix} \quad (6)$$

とすると, この中に4つの列ベクトルがあるが, これからどの2つをとっても1次独立であることが容易に確かめられる. (5)より Γ をつくると **表4** のようになる. これが強さ2の直交実験となることは(3)のような頻度を調べることによって容易に確かめられる.

特性値の構造式の中にすべての2因子交互作用効果が考えられる場合は, 実は強さ4の直交実験をつくればよいことがわかる. その場合は(5)のAとして, どの4つの列ベクトルも1次独立であるようなものを選べば(5)の Γ が強さ4の直交実験となることがわかる. また2因子交互作用の一部分だけが考えられる場合それに対応する直交実験をつくるためにガロア体上の射影幾何の点と直線の関係が利用される[3].

表 4

t_1	t_2	x_1	x_2	x_3	x_4
0	0	0	0	0	0
0	1	0	1	1	2
0	2	0	2	2	1
1	0	1	0	1	1
1	1	1	1	2	0
1	2	1	2	0	2
2	0	2	0	2	2
2	1	2	1	0	1
2	2	2	2	1	0

3. 誤り訂正コードの構成

地点 A にある情報源から1秒に1ビットの割合で情報が出現しているとする. つまり0, 1のどれかが1秒に1個の割で出現している. 簡単のため0, 1は等確率で独立に出現しているとしよう.

この情報を地点 B に送信したいのだが, 送信能力は1秒に2ビット, つまり情報の出現速度の2倍であるとする. また送信途上でノイズのため誤る確率, つまり0(1)を送って1(0)が受信される確率が p であるとする. 正しく受信される確率は $q=1-p$ である.

この誤りの確率を少なくするため, 2倍の送信能力を利用しようとしてつぎのように考えた人がいる. 情報源から0が出たら00を, 1が出たら11を送信する. つまり駄目押しのために2回送信するわけである.

情報源　0 1 1 0 …
送　信　00 11 11 00 …　(7)

こうしておいて受信側では00が受信されれば0, 11が受信されれば1が送信されたと推定し, 01あるいは10が受信されたら確率1/2で0か1かを決めるとでもしよう. こうしたら正しく受信される確率は多少増加するだろうか. 答は否なのである. いくら駄目押しをしてもこのようなやり方では少しも改善にならない.(各自確かめてみてくだ

2) したがって水準数 s が素数の累乗でないと以下の論法は成り立たない.

表 5

コード	デコード方式		
00→0000	1000	0100	0001
01→1101	0101	1001	1100
10→1010	0010	1110	1011
11→0111	1111	0011	0110

表 6

コード	デコード方式						
000→000000	100000	010000	001000	000100	000010	000001	001100
100→110100	010100	100100	111100	110000	110110	110101	111000
010→011010	111010	001010	010010	011110	011000	011011	010110
001→101001	001001	111001	100001	101101	101011	101000	100101
110→101110	001110	111110	100110	101010	101100	101111	100010
011→110011	010011	100011	111011	110111	110001	110010	111111
101→011101	111101	001101	010101	011001	011111	011100	010001
111→000111	100111	010111	001111	000011	000101	000110	001011
(000)	(100)	(010)	(001)	(110)	(011)	(101)	(111)

さい.)

ところが，情報源から出た記号をいくつかまとめて，それを適当にコード化すると，正しく受信される確率が改善されることがわかる．問題はそのコード化の方法にある．いま情報源から出る2つの記号を**表5**のようにコード化して送信するとする；

$$\begin{array}{ccccccc} 0\,1 & & 1\,0 & & 1\,1 & \cdots & \\ & \searrow & & \searrow & & \searrow & \\ & 1101 & & 1010 & & 0111 & \cdots \end{array} \qquad (8)$$

こうすると最初の2秒だけ時間おくれがあるが，あとは(8)のように連続して送信できる．

さてこの場合受信されるベクトルは16通りのものがありうるが，これを表5のようにデコードするとする．つまり表5の第1行のどの4次元ベクトルが受信されてもすべて00の出現を推定する．第2行に属するものを受けとったときは，出現記号は01であると推定するなど．

さて2つの記号をそのまま受信したときそれが2つとも正しく受信される確率は q^2 となるが，表5のようなコードをデコード方式を用いたとき正しくデコードされる確率を求めてみよう．表5でたとえば00が出現して0000が送信されたとすると，0000，1000，0100，0001が受信されたときそしてその時に限り正しくデコードされるからその確率は，

$$q^4+3pq^3 \qquad (9)$$

である．他の場もよくみると，あるコードとそれと同一行にある他の3つのベクトルとの Hamming 距離[3]が1であるから正しくデコードされる確率はすべて(9)に等しい．

$$q^4+3pq^3-q^2=pq^2(q-p) \qquad (10)$$

であり $q>p$ である限り表5による方法は正しく受信される確率を改善していることがわかる．たとえば $p=0.1$ のときには，

$$q^2=0.81, \quad q^4+3pq^3=0.8748 \qquad (11)$$

である．

さて2つをまとめて処理すると誤り受信の確率を減らすことができたが，それなら3つをまとめればさらによくなることが期待される．事実**表6**のようなコードとデコード方式をとると，正しくデコードされる確率は $P=q^6+6pq^5+p^2q^4$ で，$p=0.1$ のときは $P=0.8923$ となる．

このようにして受信の誤りの確率をいくらでも少なくできることが C. Shannon によって証明されている [4]．しかしこれを実現するためには表5,6などに示すようなコードとデコード方式を具体的につくる必要があるが，これが Hamming [4] をはじめとしてその後多くの人たちによって研究されてきた．これが誤り訂正コード問題の基本的考えである．

いままで考えた例は送信速度が情報出現速度の2倍であるという前提から，いつでも2倍の次元のベクトルへのコード化であったが，一般には情報元の n 次元ベクトルを m 次元 $(n<m)$ ベクトル

3) n 次元の0，1ベクトル $[x_1, \cdots, x_n]$，$[y_1, \cdots, y_n]$ のハミング距離とは $x_i \neq y_i$ となる i の個数．

にコード化する問題となる．誤り訂正コードの問題とはこのときどのようなコードを選びどのようなデコード方式をとるのがよいかを研究することだといえる．

すぐわかるようにコード語相互のHamming距離ができるだけ大きいコードが望ましいし，デコード方式は各コード語にHamming距離の意味でもっとも近い受信ベクトルをデコードするのがよい．

2つのベクトル $\boldsymbol{x},\boldsymbol{y}$ のHamming距離を $d(\boldsymbol{x},\boldsymbol{y})$ とし，コード V(コード語全体)の最小距離を，

$$d(V)=\min\{d(\boldsymbol{x},\boldsymbol{y}):\boldsymbol{x},\boldsymbol{y}\in V,\boldsymbol{x}\neq\boldsymbol{y}\} \quad (12)$$

で定義するとき，$d(V)$ が大きいものが望ましい．事実 $d(V)\geqq 3$ で最寄りのコード語にデコードする方式をとれば誤りが1つ起こってもそれを訂正することができるし，$d(V)\geqq 5$ ならば2つの誤りを訂正することができる．(10)で計算したような正しくデコードする確率は $d(V)$ だけからは決まらないが，いずれにせよ $d(V)$ はコード V の良さを決定的に特徴づける尺度である．

そこで m，n が与えられたとき，$d(V)$ が最大となるような V を選ぶこと，あるいは $d(V)$ が一定値以上になるような V を求めることなどはすべて組合せ論的な問題で，m，n が大きいときには大変な難問である．またデコード方式として受信ベクトルを最寄りのコード語にデコードすることも，1つ1つsearchするようでは実用にはならないので，その簡単で効率的なアルゴリズムが望まれるのである．

このような問題の多くはガロア体の導入によって一挙に解決してしまう．まず {0, 1} をGF(2)の元とみなし，この m 次元空間 $\mathrm{GF}(2)^m$ の中にコード V を選ぶことになるが，V として $\mathrm{GF}(2)^m$ の線形部分空間をとるときこれを線形コードという．線形でないコードにも良いものはあるが，構成やデコードのアルゴリズムが簡単で取り扱いやすいため現在実用化されているもののほとんどは線形コードであるといってもよい．

線形コードつまり線形空間 V を表現するには，GF(2)上 $l\times m$ 行列 H ($l=m-n$，H のランクは l)を考え，

$$V=\{\boldsymbol{x}:H\boldsymbol{x}^T=0,\ \boldsymbol{x}\in\mathrm{GF}(2)^m\} \quad (13)$$

($\boldsymbol{x}^T$ は $\boldsymbol{x}$ の転置)

とすれば V が $\mathrm{GF}(2)^m$ の中の n 次元線形部分空間になることがわかる．このような V を (m, n) —線形コードといっている．(13)における H が V を特徴づける行列でこれをパリティチェク行列とよぶ．

線形コードの利点の第1はつぎの定理にある．これによって $d(V)$ が直接求められるのである．

定理2 $d(V)=t+1$ である必要十分条件は，V のパリティチェク行列 H のどの t 列をとっても1次独立で，1次従属な $t+1$ 列が存在することである．□

この定理2によれば；たとえば $d(V)=3$ (1誤り訂正可能)のコード V をつくりたければ，どの2列も1次独立で1次従属な3列があるような行列 H をつくりさえすればよい．

例1 つぎの行列 H はどの2列も1次独立で，1次従属な3列(たとえば第1，2，4列)がある．

$$H=\begin{bmatrix}1&0&0&1&0&1\\0&1&0&1&1&0\\0&0&1&0&1&1\end{bmatrix} \quad (14)$$

したがって H をパリティチェク行列とするコード，つまり，

$$H\boldsymbol{x}^T=0:\ \begin{aligned}x_1\quad&+x_4\quad+x_6=0\\x_2\quad&+x_4+x_5\quad=0\\x_3\quad&+x_5+x_6=0\end{aligned} \quad (15)$$

の解全体 V は $d(V)=3$ であることが定理2から保証される．実は表6のコード（表6の左から第2番目の欄にあるベクトル全体）は(15)の解全体になっている．これに対して $d(V)=3$ となっていることは容易に確かめられよう．

線形コード V の表現には(13)による方法の他にもう1つ，

$$V=\{\boldsymbol{x}=\boldsymbol{t}G:\boldsymbol{t}\in\mathrm{GF}(2)^n\} \quad (16)$$

による方法がある．ここでGは GF(2) 上ランクnの$n\times m$行列で，

$$HG^T=0 \tag{17}$$

を満たすものであれば(16)が(13)と同一のものとなることは容易に確かめられよう．この表現(16)はそのまま送信アルゴリズムを与えるといってよい．情報源からn次元ベクトル$\boldsymbol{t}$が出たら，$\boldsymbol{x}=\boldsymbol{t}G$なる$m$次元ベクトルを送信すればよいのである．

最後にデコードアルゴリズム，つまり受信されたベクトルをどのコード語に対応させるかの手順を述べておこう．これもガロア体上の線形コードならきわめて簡単に処理できる．

まずVは GF(2)^{m}の線形部分空間だから，Vの平行移動によって GF(2)^{m}全体を尽すことができる．つまり $V_i=V+\boldsymbol{y}_i=\{\boldsymbol{x}+\boldsymbol{y}_i : \boldsymbol{x}\in V\}$ とすると，

$$\mathrm{GF}(2)^m=V\cup V_1\cup\cdots\cup V_M \tag{18}$$

このとき，各V_iをコセットといい，$\boldsymbol{y}_i$をコセットリーダという．(18)をコセット分割というが，各V_iのコセットリーダとしてV_iのどんな元を選んでも，この分割は不変であることが知られている．

V_iの中のウェイト（成分中の1の数）最小のものをコセットリーダとして選んで，GF(2)^{m}の元全体を**表7**のように並べたものをデコード表という．(表6は表7の具体例である．)この表で受信ベクトル$\boldsymbol{x}$が$\boldsymbol{x}_j$の行に属していれば$\boldsymbol{x}_j$にデコードするというのがデコードの原則である．

この原則を実現するのにテーブルサーチを行なうようでは意味がないが，つぎのような計算で可能である．

表 7

V	V_1	$\cdots$	V_M
$\boldsymbol{x}_1=0$	$\boldsymbol{y}_1$	$\cdots$	$\boldsymbol{y}_M$
$\boldsymbol{x}_2$	$\boldsymbol{x}_2+\boldsymbol{y}_1$	$\cdots$	$\boldsymbol{x}_2+\boldsymbol{y}_M$
$\vdots$	$\vdots$		$\vdots$
$\boldsymbol{x}_N$	$\boldsymbol{x}_N+\boldsymbol{y}_1$	$\cdots$	$\boldsymbol{x}_N+\boldsymbol{y}_M$
0	$\boldsymbol{\sigma}_1$	$\cdots$	$\boldsymbol{\sigma}_M$

受信ベクトル$\boldsymbol{x}$に対して，

$$\boldsymbol{\sigma}=\boldsymbol{x}H^T \tag{19}$$

をシンドロームというが，すぐわかるように，同一コセットの元はシンドロームが等しい．コセットリーダ$\boldsymbol{y}_i$のコセットの元はすべて同一のシンドローム，

$$\boldsymbol{\sigma}_i=\boldsymbol{y}_iH^T \tag{20}$$

をもつ．またシンドローム全体は明らかにl次元ベクトル全体に一致する（表6の( )の元はシンドローム）から，かりにこれを2進法l桁の数とみて大きさの順に並べておくこともできる．

以上の性質を利用してデコード方式の手順を述べると；受信ベクトル$\boldsymbol{x}$についてシンドローム$\boldsymbol{\sigma}=\boldsymbol{x}H^T$を計算する．$\boldsymbol{\sigma}=\boldsymbol{\sigma}_i$なら$\boldsymbol{x}$を，

$$\boldsymbol{x}_j=\boldsymbol{x}-\boldsymbol{y}_i \tag{21}$$

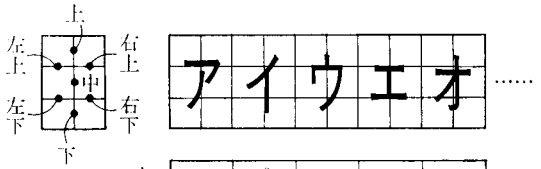
にデコードする．(以上の手順では$\boldsymbol{\sigma}_i$に対するコセットリーダ$\boldsymbol{y}_i$を記憶しておきさえすれば，テーブルサーチはいっさい不必要である)．

以上誤り訂正コードの組合せ論的難問がガロア体の導入による線形コードの利用によってうまく処理できたことをみたが，定理2に示した，どのt列も1次独立であるHをつくることも与えられm，nが大きいときは決して容易ではないし，また送信やデコードアルゴリズムの簡単な回路による実現などのため，さらにガロア体上の多項式環の理論にもとづいた巡回コードの研究などに発展している[3][4][5]．

4. ディジタル情報処理のためのガロア多項式

直交実験や誤り訂正コードの構成は数学的には GF(2) 上のn次元ベクトル空間からm次元ベクトル空間への写像という形で表現することができたが，一般にディジタル情報処理とよばれるものは，有限集合$\Omega=\{a_1, \cdots, a_N\}$から有限集合$\Gamma=\{b_1, \cdots, b_M\}$への写像$f$によって表現できるといえる．

$$\Omega \xrightarrow{f} \Gamma \tag{22}$$



上	1	0	1	0	0
左上	1	0	1	1	0
右上	1	1	1	1	1
中	1	1	0	1	1
左下	1	0	0	1	1
右下	0	1	1	1	1
下	0	1	1	0	1

図 2

例 2　文字読み取り：たとえば**図 2** のようにあるわくの中に文字が書かれたときそれを読みとるという問題を考えよう．わくの中にたとえば図 2 のように，上，左上，右上，中，…などの 7 本の線を考え，書かれた文字が線を横切れば 1，そうでなければ 0 とすると，各文字に 7 次元の 0，1 ベクトルが対応する．

実際にいろいろな人によって書かれる文字は図 2 に示すような標準的なものからは多少ずれたものとなるであろうから，文字読み取りの問題というのは**図 3** のような写像であるといえよう．この場合 $\Omega=\{0, 1\}^7$，$\Gamma=\{ア, イ, \cdots, ン\}$ と考えられる．

図 2 で考えた線の入れ方は筆者がたまたま恣意的に考えたものであるが，図 3 における各文字の原像が重くならないようにしたりまた Ω の中のブランク（どの文字にも写像されない点）があまり多くならないようにするには，わくの中にどんな線を考えるべきかなど実際にはいろいろ研究する問題は多いと思われる．□

さて (22) の関数 f を実際の装置で実現するには，磁化されているか否か，光が当っているか否かなど binary な処理媒体が用いられるから，結局のところ，a_i や b_j を 0, 1 にコード化して考えるのが妥当である．いま a_i のおのおのを 0, 1 の n 次元ベクトルに，b_j を 0, 1 の m 次元ベクトルにコード化したとすると (22) は，

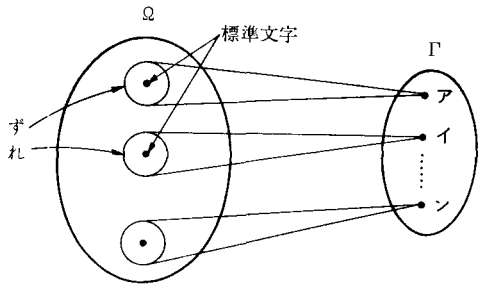


図 3

$$\begin{aligned} y_1&=f_1(x_1, \cdots, x_n)\\ &\vdots \qquad\qquad x_i, y_j\in\{0, 1\}\\ y_m&=f_m(x_1, \cdots, x_n) \end{aligned} \tag{23}$$

のように 0, 1 の n 次元ベクトル $(x_1, \cdots, x_n)$ を 0, 1 の m 次元ベクトルに写像することだといえる．

$\{0, 1\}$ を GF(2) とみなせば，(23) は結局 $GF(2)^n$ から $GF(2)^m$ への写像であるが，直交実験やコードの構成の場合は，これが線形写像であることが多かったが，一般のディジタル情報処理では線形ではせますぎる．ところがガロア体の場合はどんな関数も多項式であらわせることが示される．

定理 3　GF(2) 上の任意の n 変数関数

$$y=f(x_1, \cdots, x_n)\quad y, x_i\in GF(2) \tag{24}$$

はすべて GF(2) 上の n 次多項式でつぎのようにあらわせる（簡単のため $n=4$ について書く）．

$$f(x_1, \cdots, x_4)=\sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1}\sum_{l=0}^{1} a_{ijkl}\, x_1{}^i x_2{}^j x_3{}^k x_4{}^l \tag{25}$$

ここで a_{ijkl} はつぎのように決まる．

$$\left.\begin{aligned} &a_{0000}=f(0,0,0,0) \quad a_{1111}=\sum_x\sum_y\sum_z\sum_u f(x,y,z,u)\\ &a_{1000}=\sum_x f(x,0,0,0) \quad a_{0111}=\sum_x\sum_y\sum_z f(0,x,y,z)\\ &a_{0100}=\sum f(0,x,0,0) \quad a_{1011}=\sum\sum\sum f(x,0,y,z)\\ &a_{0010}=\sum f(0,0,x,0) \quad a_{1101}=\sum\sum\sum f(x,y,0,z)\\ &a_{0001}=\sum f(0,0,0,x) \quad a_{1110}=\sum\sum\sum f(x,y,z,0)\\ &a_{1100}=\sum_x\sum_y f(x,y,0,0) \quad a_{0011}=\sum_x\sum_y f(0,0,x,y)\\ &a_{1010}=\sum\sum f(x,0,y,0) \quad a_{0101}=\sum\sum f(0,x,0,y)\\ &a_{1001}=\sum\sum f(x,0,0,y) \quad a_{0110}=\sum\sum f(0,x,y,0) \end{aligned}\right\} \tag{26}$$

ここで $\sum_x, \sum_x\sum_y$ などの x, y はそれぞれ GF(2)$=\{0,1\}$ の元全体について動くものとする．□

この定理の証明はそれほど難かしくはないが，未定係数法あるいはフーリエ展開と同様な原理にもとづいている．

例として**表8**のような変換が与えられたとするとき，これから(23)のような関数をつくってみよう．定理1を各 y_i について適用すると，

$$\left.\begin{aligned} y_1 &= x_1x_2+x_2x_3+x_3x_4+ \\ &\quad x_4x_1+x_1x_3+x_2x_4 \\ y_2 &= x_1+x_2+x_3+x_4 \end{aligned}\right\} \tag{27}$$

となる．

表 8

x_1	x_2	x_3	x_4	y_1	y_2
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	1	0
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	0	0

表8は各 y_i についてみればいわゆる真理表であるから，ブール代数によって関数(23)を表現することもできる．ブール代数ではいわゆる標準形を求め，必要ならそれを簡単化して(27)に相当する式を導き出すのである．ガロア体によれば(26)のような公式1本で結果が出るという意味でガロアびいきである筆者にはすっきりしているように思えるが，客観的にみるとここまでは本質的な差はないといえよう．

ところが，ガロア体の拡大という概念を利用すると(23)のように n 変数の多項式で m 個つくるのでなく，これらを1変数の1つの多項式にまとめ上げてしまうことができるのである［6］．拡大という概念は体論固有のものであるため，このような発想はブール代数の中には生まれてこなかったのである．

例 3　ハッシング，キーワード縮約

ハッシングというのは，たとえば部品コードとか学生番号など人間にとって意味のわかりやすいコード化をすると，その桁が多くなり過ぎるため，コンピュータの中でそれをコンパクトな情報に縮約しようという考えである．

これとほぼ同様の考えは情報検索におけるキーワードの場合にも適用できる．たとえば文献を特徴づけるキーワードが n 字あって，それらを $w_1, w_2, \cdots, w_n$ とすると，ある文献が w_i をもてば第 i 成分が1，そうでなければ0であるような n 次元0，1ベクトルでその文献を表現することができる．

ところが実際(たとえば図書館など)に存在する文献の総数を N とすると，$N \ll 2^n$ となるのが常である．そこで実際の文献(を表現するベクトル)は n 次元空間の中にきわめて疎に散在しているに過ぎない．そこで $N \fallingdotseq 2^m$($N \leqq 2^m$ である必要はあるが)なる m を選んで，n 次元ベクトルから m 次元ベクトルへの写像を考え，コンピュータの中では縮約された m 次元コードで，検索などの処理を行なおうという考えが生まれる．この考えをキーワード縮約とよぶことにしよう．

ハッシング(必要ならばbinaryのコード化をして)やキーワード縮約は $\Omega=$GF(2)^n から $\Gamma=$GF(2)^m への写像として，われわれの一般形(22)，(23)に適合する．しかしこの場合注意すべきことは，GF(2)^n の元全体を GF(2)^m に写像するのではなく，GF(2)^n の中で文献が存在している点だけについて写像を考えればよいことになる．

文献の存在している点全体を R，それ以外の点全体を $\bar{R}=$GF(2)^n-R として置くと，$\bar{R}$ の中の点に対する写像はまったく自由であるから以上の解析の中でこの自由性を利用することができる．

たとえば多項式(25)の高次の項を切り捨てる(ちょうど要因配置実験で高次の交互作用効果を無視するように)といったことが可能である．しかしそのときはもはや(26)の公式は使えない．未定係数法の原理にしたがってGF(2)上の連立一次方程式を解くことによって係数 a_{ijkl} を決めねばならないだろう．

(25)(26)の公式をそのまま使うためには，$\bar{R}$ の点はすべてGF(2)^m の原点0に写像するという約

束にしておくと，(26)で $\sum_x, \sum_x\sum_y$ の意味を x, (x,y)が R の所だけを動いて加えるということにしておくと以上の解析はそっくりそのままでよいことになる．

キーワード縮約のもう一つの特徴は，これが多くの桁を少ない桁に縮約することだけが目的なのだから，各文献の表現ベクトルを $GF(2)^m$ のどの点に写像するかはまったく自由に選べるということである．この意味での自由性はどのように利用したらもっとも良いかといった決め手はいまのところ見当らない．

さてハッシングやキーワード縮約のような情報圧縮の原理は，人間に解りやすい冗長度のある情報をコンピュータで処理しやすいコンパクトな情報に圧縮することであるが，コンピュータで処理した後で人間の言葉に戻す必要，つまり逆変換も必要である．(従来のハッシング法はこの逆変換が不可能であった)．

この逆写像をつくるためにはいままでの $GF(2)^n$ と $GF(2)^m$ の役割をかえさえすればよい．つまりハッシングやキーワード縮約のためには正変換用の関数 f と逆変換用の関数 $\bar{f}$ とを(25)(26)の方法で別々につくっておけばよいのである．

例4　漢字プリント：例3の逆変換でみたような小さな空間から大きな空間への写像は，漢字プリントの問題にも有用な方法を与える．

漢字をたとえば50×50の細分されたマス目の白黒で印刷するためには，50×50＝2500ビットの情報で漢字を記憶しておかねばならない．漢字はせいぜい10000($\leqq 2^{15}$)だから，15ビットもあれば記憶できるはずである．したがって $GF(2)^{15}$ から $GF(2)^{2500}$ への写像関数を(25)(26)にならってつくれば良いことになる．

参考文献

［1］ 奥野忠一，芳賀敏郎：実験計画法，培風館 昭和44年9月．

［2］ S. Moriguti : Optimality of orthogonal designs, *Rep. Stat. Appl. Res.* JUSE 3(1954).

［3］ 高橋磐郎：組合せ理論，岩波（近刊）．

［4］ J. H. van Lint : *Coding Theory*, Springer (1971).

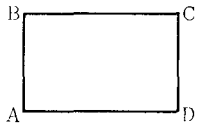
［5］ W. W. Peterson & E. J. Weldon Jr. : *Error Correcting Codes*, 2nd ed. MIT Press (1972).

［6］ 高橋磐郎：ディジタル情報処理へのガロア体の応用，数理科学　1978年8月．

フォーラム

数理パズルを楽しもう (10)

問題　勝手な形の長方形ABCDがあります．これを2本の直線で3片に切り分け，うまく組み替えて正方形にしたいのです．どのようにすれば，よいでしょうか．ただし，長方形の横の長さは，縦の長さの4倍よりは短いものとします．

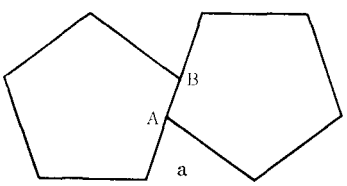


〔7月号(440ページ)の解答〕同じ大きさの2つの正五角形を，図aのような配置にかくのがポイントである．すると，2辺の重なり合った部分として，線分ABができるが，これを1辺の長さとする12個の正五角形が，簡単な作図で図bのようにできる．これから，図cのような切り抜きをつくれば，それが求める展開図である．この作図の証明は初等的なので，省略させていただく．

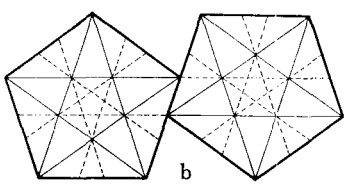
なお，この問題は高木貞治先生の著書〔1〕からヒントを得たものである．

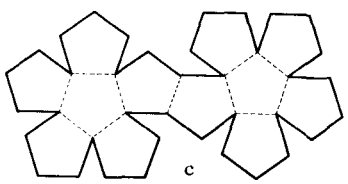
〔1〕高木貞治，数学小景，岩波書店，1943．

（中村義作　信州大学工学部）



a


b


c

FORUM