2011 年度計算機数学 B (ver. 1.8)

東北大学理学部数学科　石田正典

序文

この講義では多項式および多項式環のイデアルの計算を通じて計算機の数学への応用を考えて行きたい．使う言語は C と Singular（文献 [1] 参照）を用いる予定である．

多項式を与えるデータは変数の文字と係数の列である．C にはデータ型として配列があり，0 番目からの有限数列を定義できる．多項式の変数の文字が例えば x と固定されている場合，数値の配列を 0 次からの係数の列がその配列となる多項式を表しているものと考えることが出来る．C では多次元配列も使えるので，これは同様に多変数多項式と考えられる．

第 1 節と第 2 節では多項式を配列で書いてみることにより多項式計算の仕組みを確かめる．その後で終結式，判別式など多項式の古典的な結果について紹介する．これらは式が大きくなりすぎる欠点があったが，計算機の能力の向上やグレブナー基底の理論の発展と共に復活し重要性を増している．

具体的な環のイデアルの計算にはグレブナー基底の理論が大きな力を発揮することが示されている．グレブナー基底を使える計算ソフトはいくつかあるが，この講義では Singular を使って実際の計算を行う．また，Maple も利用出来るので紹介する予定である．

1　多項式

代数学で扱う多項式は変数としての記号 x と係数 $a_0, \ldots, a_m$ により

$$f(x) = a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0$$

と表される式のことである．係数は一般にはある四則演算のできる集合 K の元をとる．この K を係数体というが，当面 K を 有理数全体 $\mathbf{Q}$ または実数全体 $\mathbf{R}$ とする．すなわち，$a_0, \ldots, a_m$ は有理数か実数である．

多項式は中学校から学んでいるので常識的になって気づきにくいが，書き方に次のルールがあることに注意する．

(1) 係数が 0 の項は省略する．例えば $0x^3 + 2x^2 + 0x + 5$ は通常 $2x^2 + 5$ と書く．ただし，係数が文字定数を含む場合は 0 になっても構わない．例えば $2ax + 1$ は $a = 0$ の場合もあり得る．なお，係数が全部 0 の多項式は 0 と書く．

(2) 係数が負の場合や，ある定数を含む式の -1 倍の場合は $-$ の記号を使って表す．すなわち，$x^2+(-2)x+1=x^2-2x+1$ であり $x+-a$ とは書かず $x-a$ と書く．

上記の $f(x)$ で $a_m \neq 0$ の場合，m を $f(x)$ の次数と言い，$\deg f(x)$ で表す．0 以外の多項式には次数が定まる．0 以外の定数の次数が 0 であることに注意が必要である．多項式 0 の次数は考えないのが普通であるが，出てきた場合は次数を -1 または $-\infty$ とするのが適当である．

多項式の比較は係数により行う．すなわち，

$$(1)\quad f(x)=a_mx^m+a_{m-1}x^{m-1}+\cdots+a_1x+a_0\ ,\ \ g(x)=b_nx^n+b_{n-1}x^{n-1}+\cdots+b_1x+b_0$$

の場合，$f(x)=g(x)$ は $m\geq n$ なら $a_m=\cdots=a_{n+1}=0$ かつ $a_n=b_n,\ldots,a_0=b_0$ となることで，$m<n$ なら $b_n=\cdots=b_{m+1}=0$ かつ $a_m=b_m,\ldots,a_0=b_0$ となることである．ここで $m\geq n$ と $m<n$ に場合分けして記述が複雑になったが，$m\neq n$ ならあらかじめ係数 0 の項をどちらかに加えて $m=n$ と仮定することが出来る．その場合，等しいための条件は単に $a_m=b_m,\ldots,a_0=b_0$ である．多項式の和 $f(x)+g(x)$ は同様に高い方の次数に合わせて $m=n$ と仮定すれば，

$$(2)\qquad f(x)+g(x)=(a_m+b_m)x^m+(a_{m-1}+b_{m-1})x^{m-1}+\cdots+(a_1+b_1)x+(a_0+b_0)$$

で定義される．

例 1.1 $f(x)=2x^2-3x+1,\ g(x)=5x-3$ とすると

$$f(x)=2x^2+(-3)x+1\ ,\ \ g(x)=0x^2+5x+(-3)$$

であるから

$$f(x)+g(x)=(2+0)x^2+(-3+5)x+(1-3)=2x^2+2x-2$$

となる．もちろん，中学校以来行ってきた計算通りであるが，定義として書けば (2) のようになり，多項式計算を計算機でプログラムする場合はこれに従ってプログラムを記述する必要がある．

多項式 $f(x)=a_mx^m+a_{m-1}x^{m-1}+\cdots+a_1x+a_0$ は総和の記号を用いて $\sum_{k=0}^{m}a_kx^k$ と書くこともできる．この記述を使って，(1) で一般型を与えた 2 つの多項式の積は

$$(3)\qquad f(x)g(x)=\sum_{k=0}^{m+n}\left(\sum_{i=0}^{k}a_ib_{k-i}\right)x^k$$

で定義される．ただし，$i>\deg f$ の場合 $a_i=0$, $j>\deg g$ の場合 $b_j=0$ と考える．これも通常通り式の積を展開して整理することと同じである．ここで $\deg f(x)=m$, $\deg g(x)=n$ であれば $a_m\neq 0$, $b_n\neq 0$ であり，$\deg f(x)g(x)=m+n$ となり次の定理を得る．

定理 1.2 $f(x),g(x)$ を 0 でない多項式とすると $\deg f(x)g(x)=\deg f(x)+\deg g(x)$ となる．

例 1.3 例 1.1 と同様に $f(x) = 2x^2 - 3x + 1$, $g(x) = 5x - 3$ として (3) に従って計算すると，定数項は $1 \cdot (-3) = -3$, 1 次の項の係数は $1 \cdot 5 + (-3) \cdot (-3) = 14$, 2 次の項の係数は $1 \cdot 0 + (-3) \cdot 5 + 2 \cdot (-3) = -21$, 3 次の項の係数は $1 \cdot 0 + (-3) \cdot 0 + 2 \cdot 5 + 0 \cdot 3 = 10$ となるので，積は

$$f(x)g(x) = 10x^3 - 21x^2 + 14x - 3$$

となる．普通に積を展開して整理する計算と同じことがわかる．

整数係数の多項式の和と積について C でプログラムした例を下に示す．なお，プログラム中のコメントで「x^n」のような指数の表記に「x^n」を使っている．これはテキスト形式の冪の記述として一般的であり，数式を含む文書の作成ソフトである TeX での記述や Maple などの計算ソフトの冪乗の指定にも使われる．

```
/*
                計算機数学 B プログラム 1
                多項式の和と積
*/
#include <stdio.h>

/*  高々 15 次の多項式の係数を 0 次から順に並べる（自由に変更可） */
static int data1[16] = {1,2,1,2,1,2,0,0,0,0,0,0,0,0,0,0};
/* 2x^5 + x^4 + 2x^3 + x^2 + 2x + 1 */
static int data2[16] = {3,1,3,1,3,1,0,0,0,0,0,0,0,0,0,0};
/* x^5 + 3x^4 + x^3 + 3x^2 + x + 3 */

/*  多項式の構造体を定義する  */
struct polynom {
    char var; // 変数の文字
    int deg;  // 次数，ただし deg 0 = -1 とする．
    int coef[16]; // coef[i] が i 次の係数
};

/*  関数の宣言  */
void xpoly(struct polynom *pf);
void addpoly(struct polynom g, struct polynom h, struct polynom *pf);
void multpoly(struct polynom g, struct polynom h, struct polynom *pf);

int main () {
    int i;
    struct polynom f, g, h;
```

```
    for (i=0; i<16; i++) {
        g.coef[i] = data1[i];
        h.coef[i] = data2[i];
    }
    xpoly(&g);
    xpoly(&h);
    addpoly(g,h,&f);
    for (i=0; i<16; i++) {
        printf("%d ",f.coef[i]);
    }
    printf("\n");
    multpoly(g,h,&f);
    for (i=0; i<16; i++) {
        printf("%d ",f.coef[i]);
    }
    printf("\n");
    return 0;
}

/*  変数を x とし，次数を調べて記入する  */
void xpoly(struct polynom *pf) {
    int i;
    pf->var = 'x';
    for (i=15; i>=0; i--) {
        if (pf->coef[i]!=0) {
            pf->deg = i;
            return;
        }
    }
    pf->deg = -1;
    return;
}

/*  多項式 g, h の和を計算し，ポインタ pf を使い結果を f に入れる  */
void addpoly(struct polynom g, struct polynom h, struct polynom *pf)
{
    int i;
    if (g.var != h.var) {
        printf("error : polynomial variable are not same");
```

```
        return;
    }
    for (i=0; i<16; i++) {
        pf->coef[i] = g.coef[i] + h.coef[i];
    }
    xpoly(pf);
    return;
}

/*  多項式 g, h の積を計算し, ポインタ pf を使い結果を f に入れる  */
void multpoly(struct polynom g, struct polynom h, struct polynom *pf)
{
    int i, k, d, a, b;
    if (g.var != h.var) {
        printf("error : polynomial variable are not same");
        return;
    }
    if ((d = g.deg + h.deg) > 15) {
        printf("error : degrees are too large");
        return;
    }
    for (i=0; i<16; i++) {
        pf->coef[i] = 0;
    }
    for (k=0; k<=d; k++) {
        a = 0;
        if (k > g.deg) a = k - g.deg;
        b = k;
        if (k > h.deg) b = h.deg;
        for (i=a; i<=b; i++) {
            pf->coef[k] += g.coef[k-i] *h.coef[i];
        }
    }
    pf->deg = d;
    pf->var = 'x';
    return;
}
```

上記のプログラムでは多項式の係数列に整数の配列を使う．`coef[i]` を x^i の係数とするのが合理的である．そのため，例を与える数列 `data1`, `data2` は 0 次の係数から順に並べる．この意味では多項式を

$$f = a_0 + a_1 x + \cdots + a_{m-1}x^{m-1} + a_m x^m$$

と次数 0 から書くのも合理的である．上記のプログラムの関数 `addpoly(g,h,&f)` では多項式 `g`, `h` の和を計算して `f` として返している．最後の部分が少し複雑になっているのは，乗算の回数を減らすため係数が 0 となる部分の係数の乗算を排除したためである．例えば m, n 次の多項式の積では x^{m+n} の係数の計算は (3) の式では $k = m + n$ で乗算は $m + n + 1$ 回であるが，実際の係数の乗算は $a_m b_n$ の 1 回で済む．

例 1.4 プログラム 1 の `data1`, `data2` を

```
static int data1[16] = {-1,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
static int data2[16] = {3,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0};
```

と置き換えて実行すると

```
2 4 2 0 0 0 0 0 0 0 0 0 0 0 0 0
-3 8 1 6 0 0 0 0 0 0 0 0 0 0 0 0
```

と表示される．これは，それぞれ

$$(3x-1) + (2x^2 + x + 3) = 2x^2 + 4x + 2\ ,\ \ (3x-1)(2x^2 + x + 3) = 6x^3 + x^2 + 8x - 3$$

を示す．

多項式の変数には値を代入することができる．すなわち $f(x) = a_m x^m + a_{m-1}x^{m-1} + \cdots + a_1 x + a_0$ で c が定数なら，$f(c) = a_m c^m + a_{m-1}c^{m-1} + \cdots + a_1 c + a_0$ となる．ここで，多項式の係数を体 K からとっているとしても，K が別の体 L に入っていれば L の任意の元を代入することができる．たとえば整数係数の多項式の変数に実数を代入することができる．これは当然のことではあるが，計算機でプログラムする場合は整数から実数への型の変換に注意する必要がある．

次の定理は証明は省略するが次数についての数学的帰納法で証明できる．意味は，式で計算してから変数に数値を代入しても先に数値を代入してから計算しても結果は同じという誰もが当然としていることである．代数の講義ではこれは「L の元の代入は多項式環 $K[x]$ から L への準同型写像となる」という言い方になる．ただし，計算機で実数計算を行う場合は必然的に近似計算となるので，変数に数値をいつ代入するかで誤差を生じる場合もある．

定理 1.5 $g(x), h(x)$ 多項式とすると，多項式 $f(x) = g(x)h(x)$ と任意の定数 c について，$f(c) = g(c) + h(c)$ および $f(c) = g(c)h(c)$ が成り立つ．

問 1.6 整数係数の多項式の変数への実数の代入を計算するプログラムを作り，いくつかの例について定理 1.5 を確かめよ．なお，誤差が出たとしてもプログラムミスと考える必要はない．

多項式について一般的な議論をするためには多項式全体の集合を考える必要がある．すなわち，係数をとる体を K として x を変数とする多項式全体を $K[x]$ と書く．集合 $K[x]$ の元は多項式となるが，必ずしも $f(x)$ のように書かずに単に f と書くことも多い．$f(x)$ と書いても f と書いても同じ多項式である．

2 元 $f, g \in K[x]$ に対して前述のように和 $f+g$ と積 fg が定義される．この和と積について集合 $K[x]$ は可換環となる．可換環の定義，すなわち可換環 A が必ず持っている性質は次の通りである．

(1) 任意の $x, y, z \in A$ について，和と積それぞれの結合法則

$$(x+y)+z = x+(y+z)\ ,\ \ (xy)z = x(yz)$$

が成り立つ．

(2) A には和の単位元 0 と積の単位元 1 が存在して，任意の $x \in A$ について

$$0+x = x+0 = x\ ,\ \ 1x = x1 = x$$

が成り立つ．

(3) 任意の $x \in A$ に対して和の逆元 $x' \in A$ が存在して $x+x' = x'+x = 0$ となる．

(4) 任意の $x, y \in A$ に対して交換法則 $x+y = y+x$ および $xy = yx$ が成り立つ．

(5) 任意の $x, y, z \in A$ について分配法則

$$(x+y)z = xz+yz\ ,\ \ x(y+z) = xy+xz$$

が成り立つ．

加法について群をなすという加群の定義を知っていれば，A が加群で，乗法について結合法則を満たし単位元 1 を持ち，分配法則が成り立つのが環で，さらに乗法について交換法則が成り立てば可換環である．

$K[x]$ を多項式環と言う．一般の可換環では x, y が 0 でなくても $xy = 0$ となり得るが，多項式環の場合は $f, g \neq 0$ なら $fg \neq 0$ である．実際，$f, g \in K[x] \setminus \{0\}$ で $m = \deg f$, $n = \deg g$ とすれば定理 1.3 により fg は $m+n$ 次の 0 でない多項式である．

2　多変数多項式

x, y を変数とする 2 変数多項式，x, y, z を変数とする 3 変数多項式が考えられる．例えば，x^2+y^2-1 は 2 変数多項式で $x^3+y^3+z^3-3xyz$ は 3 変数多項式である．

一般には n を正の整数とし，$x_1, \ldots, x_n$ を変数とする多項式を考えることができる．$i_1, \ldots, i_n$ を負でない整数とするとき $x_1^{i_1} \cdots x_n^{i_n}$ の形の式を単項式と言う．$(i_1, \ldots, i_n)$ をこの単項式の指

数ベクトルまたは単に指数という．単項式に係数をつけて足し合わせたものが n 変数多項式である．多項式を記述するとき，第 1 節で扱った 1 変数多項式の場合は高い次数の項からか，あるいは低い次数の項から並べれば良いが，多変数の場合は標準的な書き方は決めにくい．単項式全体に全順序を入れることにより多項式の書き方を統一することができるが，順序の入れ方については第 4 節で述べる．総和の記号を使えば，$\mathbf{Z}_0$ を 0 以上の整数全体として，n 変数多項式の一般形を

$$f = \sum_{(i_1,\ldots,i_n)\in\mathbf{Z}_0^n} a_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n}$$

と書くことが出来る．ただし，有限個の $(i_1,\ldots,i_n)$ を除いては $a_{i_1,\ldots,i_n} = 0$ とする．

2 変数多項式の計算のプログラムを考える．x, y の単項式 $x^i y^j$ の指数ベクトル (i, j) は 2 つの非負整数の組である．ここである整数 $N > 0$ を固定して x, y について高々 $N - 1$ 次の多項式を考えるとすると，$0 \le i, j < N$ となるので，多項式の係数 $a_{i,j}$ のデータは C では 2 次元配列 `a[N][N]` で表すことができる．すなわち，配列の要素 `a[i][j]` が $x^i y^j$ の係数 $a_{i,j}$ である．例えば，$N = 4$ で整数係数の多項式であれば

```
int a[4][4] =
   {{7,5,3,1},
    {4,0,0,0},
    {1,0,0,0},
    {0,0,8,0}};
```

として多項式

$$f(x, y) = 7 + 5y + 3y^2 + y^3 + 4x + x^2 + 8x^3y^2$$

の係数のデータを表すことが出来る．さらに言えばこの配列をプログラムの中での多項式と考えることが出来る．

第 1 節のプログラム 1 を 2 変数用に書き換えると次のようになる．

```
/*
                poly2.c
                計算機数学 B プログラム 2
                2 変数多項式の和と積
*/
#include <stdio.h>

/*  高々 7 次の 2 変数多項式の係数を 0 次から辞書式順序で並べる  */
static int data1[8][8] =
    {{1,2,3,4,0,0,0,0},
    {1,2,3,4,0,0,0,0},
    {1,2,3,4,0,0,0,0},
```

```
    {1,2,3,4,0,0,0,0},
    {0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,0}};
static int data2[8][8] =
    {{1,2,3,4,0,0,0,0},
    {1,2,3,4,0,0,0,0},
    {1,2,3,4,0,0,0,0},
    {1,2,3,4,0,0,0,0},
    {0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,0}};

/*  2 変数多項式の構造体を定義する  */
struct polyn2 {
    char var[2]; // 変数の文字
    int deg[2];  // 次数, ただし deg 0 = (-1, -1) とする.
    int coef[8][8]; // coef[i][j] が (i, j) 次の係数
};

/*  関数の宣言  */
void xypoly(struct polyn2 *pf);
void addpoly2(struct polyn2 g, struct polyn2 h, struct polyn2 *pf);
void multpoly2(struct polyn2 g, struct polyn2 h, struct polyn2 *pf);


int main () {
    int i, j;
    struct polyn2 f, g, h;
    for (i=0; i<8; i++) {
        for (j=0; j<8; j++) {
            g.coef[i][j] = data1[i][j];
            h.coef[i][j] = data2[i][j];
        }
    }
    xypoly(&g);
    xypoly(&h);
```

```
    addpoly2(g,h,&f);
    for (i=0; i<8; i++) {
        for (j=0; j<8; j++) {
            printf("%d ",f.coef[i][j]);
        }
        printf("\n");
    }
    printf("\n\n");
    multpoly2(g,h,&f);
    for (i=0; i<8; i++) {
        for (j=0; j<8; j++) {
            printf("%d ",f.coef[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    return 0;
}

/*  変数を x とし，次数を調べて記入する  */
void xypoly(struct polyn2 *pf) {
    int i, j, d1=-1, d2=-1;
    pf->var[0] = 'x';
    pf->var[1] = 'y';
    for (i=0; i<8; i++) {
        for (j=0; j<8; j++) {
            if (pf->coef[i][j]!=0) {
                if (i > d1) d1 = i;
                if (j > d2) d2 = j;
            }
        }
    }
    pf->deg[0] = d1;
    pf->deg[1] = d2;
    return;
}

/*  多項式 g, h の和を計算し，ポインタ pf を使い結果を f に入れる  */
void addpoly2(struct polyn2 g, struct polyn2 h, struct polyn2 *pf)
```

```
{
    int i, j;
    if (g.var[0] != h.var[0] || g.var[1] != h.var[1]) {
        printf("error : polynomial variables are not same");
        return;
    }
    for (i=0; i<8; i++) {
        for (j=0; j<8; j++) {
        pf->coef[i][j] = g.coef[i][j] + h.coef[i][j];
        }
    }
    xypoly(pf);
    return;
}

/*  多項式 g, h の積を計算し, ポインタ pf を使い結果を f に入れる  */
void multpoly2(struct polyn2 g, struct polyn2 h, struct polyn2 *pf)
{
    int i, j, k1, k2, d1, d2, a1, a2, b1, b2;
    if (g.var[0]!=h.var[0] || g.var[1]!=h.var[1]) {
        printf("error : polynomial variables are not same");
        return;
    }
    if ((d1=g.deg[0]+h.deg[0])>7 || (d2=g.deg[1]+h.deg[1])>7) {
        printf("error : degrees are too large");
        return;
    }
    for (i=0; i<8; i++) {
        for (j=0; j<8; j++) {
            pf->coef[i][j]=0;
        }
    }
    for (k1=0; k1<=d1; k1++) {
        a1 = 0;
        if (k1 > g.deg[0]) a1 = k1 - g.deg[0];
        b1 = k1;
        if (k1 > h.deg[0]) b1 = h.deg[0];
        for (k2=0; k2<=d2; k2++) {
            a2 = 0;
```

```
            if (k2 > g.deg[1]) a2 = k2 - g.deg[1];
            b2 = k2;
            if (k2 > h.deg[1]) b2 = h.deg[1];
            for (i=a1; i<=b1; i++) {
                for (j=a2; j<=b2; j++) {
                    pf->coef[k1][k2] += g.coef[k1-i][k2-j]*h.coef[i][j];
                }
            }
        }
    }
    pf->deg[0] = d1;
    pf->deg[1] = d2;
    pf->var[0] = 'x';
    pf->var[1] = 'y';
    return;
}
```

変数 $x_1, \ldots, x_n$ についての多項式全体からなる集合 $K[x_1, \ldots, x_n]$ も可換環となり，これを n 変数多項式環と言う．これは次のように考えることも出来る．第 1 節で体 K 上の多項式環 $K[x]$ を定義したが，任意の可換環 R 上の多項式環 $R[x]$ も R の元を係数とする多項式全体からなる環として定義できる．ここで $R = K[x]$ として変数 y の多項式環 $(K[x])[y]$ を考えると，これが 2 変数多項式環 $K[x, y]$ である．さらに $K[x, y][z]$ が $K[x, y, z]$ である．一般には

$$K[x_1, \ldots, x_n] = K[x_1, \ldots, x_{n-1}][x_n]$$

と $K[x_1]$ から $n = 2, 3, \ldots$ と順に定義される．このような定義を帰納的定義といい，一般的定理などを証明するときに数学的帰納法が使える形になっている．

3 終結式

多項式の係数を与える体 K を固定する．0 でない多項式 $f(x)$ は 1 次以上の 2 つの多項式 $p(x), q(x)$ により $f(x) = p(x)q(x)$ と書けるとき可約，書けないとき既約という．多項式環での既約多項式は整数全体における素数と同様の意味を持ち，整数の素因数分解と同様のことができる．すなわち，任意の多項式 $f(x)$ は既約であるか，または 2 つ以上有限個の既約多項式 $p_1(x), \ldots, p_s(x)$ により $f(x) = p_1(x) \cdots p_s(x)$ と因数分解できる．$p_1(x), \ldots, p_s(x)$ を $f(x)$ の既約因子という．この分解は順序と各項に K の 0 でない元をかけることを除いて一意的である．このことは可換環論では体上の任意変数の多項式環は一意分解環となるという一般的定理の 1 変数多項式環の場合である．この定理の証明に至る既約多項式の重要な性質として，「$f(x)$ が既約で $f(x)|g(x)h(x)$ であれば $f(x)|g(x)$ または $f(x)|h(x)$」がある．ここで $p(x)|q(x)$ は $q(x)$ が $p(x)$ で割り切れるという記号である．

注意 3.1 最高次の係数が 1 の多項式をモニックな多項式と言う．0 でない任意の多項式は最高次の係数の逆元をかけるとモニックな多項式になる．モニックな多項式で割る計算は記述しやすい．実際，$f(x)$ を m 次のモニックな多項式として，多項式 $g(x)$ の次数 n が m 以上で x^n の係数が a であれば $\deg\{g(x) - ax^{n-m}f(x)\} < n$ となる．これを繰り返して次数を $m-1$ 以下にすることにより $g(x)$ を $f(x)$ で割った商と余りが得られる．

0 でない 2 つの多項式 $f(x)$ と $g(x)$ が共通因子を持つか否かの問題を考える．
$\deg f(x) = m$, $\deg g(x) = n$ で

$$f(x) = a_m x^m + a_{m-1} + \cdots + a_1 x + a_0\ ,\quad g(x) = b_n x^n + b_{n-1}x^{n-1} + \cdots + b_1 x + b_0$$

とする．

命題 3.2 m 次と n 次の多項式 $f(x), g(x)$ が共通因子を持つための必要十分条件は，$n-1$ 次と $m-1$ 次以下の 0 でない多項式 $p(x), q(x)$ が存在して $p(x)f(x) + q(x)g(x) = 0$ となることである．

証明　$f(x), g(x)$ が共通因子 $h(x)$ を持つとする．$\deg h(x) = l$ とすると $m-l$ 次と $n-l$ 次の多項式 $q(x), r(x)$ が存在して $f(x) = q(x)h(x)$, $g(x) = r(x)h(x)$ となる．このとき $p(x) = -r(x)$ と置けば

$$\begin{aligned} & p(x)f(x) + q(x)g(x) \\ = \ & p(x)q(x)h(x) + q(x)r(x)h(x) \\ = \ & -r(x)q(x)h(x) + q(x)r(x)h(x) \\ = \ & 0 \end{aligned}$$

となる．$l > 0$ であるから条件を満たす．
逆にこのような $p(x), q(x)$ が存在するとすれば $p(x)f(x) = -q(x)g(x)$ となる．両辺を既約多項式の積に分解すると，分解の一意性から $f(x)$ の因子はすべて重複度も込めて右辺にも現れるが．次数の和は m であるから高々 $m-1$ 次の $q(x)$ の因子だけでは不足する．したがって $g(x)$ は $f(x)$ と共通因子をもつ．　証明終わり

この命題は次に述べるようにベクトル空間での一次独立性の条件に書き直すことが出来て，線形代数の定理を適用することが可能になる．
$f(x)$ と $g(x)$ はこれまで同様に m 次と n 次の多項式とする．$m+n-1$ 次以下の多項式全体に 0 を加えた集合 V は $\{x^{m+n-1}, x^{m+n-2}, \ldots, x, 1\}$ を基底とする $m+n$ 次元ベクトル空間である．V の n 個の元

$$x^{n-1}f(x), x^{n-2}f(x), \ldots, xf(x), f(x) \tag{4}$$

と m 個の元

$$x^{m-1}g(x), x^{m-2}g(x), \ldots, xg(x), g(x) \tag{5}$$

の合計 $m+n$ 個を考える．$p(x)=p_{n-1}x^{n-1}+\cdots+p_1x+p_0$, $q(x)=q_{m-1}x^{m-1}+\cdots+q_1x+q_0$ とすると，

$$p(x)f(x)=p_{n-1}x^{n-1}f(x)+\cdots+p_1xf(x)+p_0f(x)\ ,$$
$$q(x)f(x)=q_{m-1}x^{m-1}g(x)+\cdots+q_1xg(x)+q_0g(x)$$

である．この 2 つの等式の左辺も右辺の各項も $m+n-1$ 次以下の多項式なので，命題 3.2 の等式 $p(x)f(x)+q(x)g(x)=0$ はベクトル空間 V での等式

$$p_{n-1}x^{n-1}f(x)+\cdots+p_1xf(x)+p_0f(x)+q_{m-1}x^{m-1}g(x)+\cdots+q_1xg(x)+q_0g(x)=0$$

となる．$p(x),q(x)\neq 0$ とすれば，これは V の (4) と (5) の $m+n$ 個の元が一次従属であることを示す．逆にこれらが一次従属であれば，命題 3.1 の条件を満たす 0 でない多項式 $p(x),q(x)$ が存在する．従って，命題 3.2 により $f(x),g(x)$ が共通因子を持つための必要十分条件は これら $m+n$ 個の元が一次従属であることとなる．

基底 $\{x^{m+n-1},x^{m+n-2},\ldots,x,1\}$ について (4) と (5) の元を縦ベクトルとして書いた $m+n$ 次正方行列を R とすると

$$R=\begin{bmatrix}
a_m & 0 & 0 & \cdots & 0 & b_n & 0 & 0 & \cdots & 0\\
a_{m-1} & a_m & 0 & & \vdots & b_{n-1} & b_n & 0 & & \vdots\\
\vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots\\
\vdots & \ddots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & \ddots & 0\\
\vdots & \ddots & \ddots & \ddots & a_m & \vdots & \ddots & \ddots & \ddots & b_n\\
\vdots & \ddots & \ddots & \ddots & a_{m-1} & \vdots & \ddots & \ddots & \ddots & b_{n-1}\\
a_0 & a_1 & a_2 & \ddots & \vdots & b_0 & b_1 & b_2 & \ddots & \vdots\\
0 & a_0 & a_1 & \ddots & \vdots & 0 & b_0 & b_1 & \ddots & \vdots\\
\vdots & 0 & \ddots & \ddots & \vdots & \vdots & 0 & \ddots & \ddots & \vdots\\
\vdots & & \ddots & \ddots & a_1 & \vdots & & \ddots & \ddots & b_1\\
0 & \cdots & \cdots & 0 & a_0 & 0 & \cdots & \cdots & 0 & b_0
\end{bmatrix}$$

となる．ただし，行を見る場合は左の n 列と右の m 列は分けて見て欲しい．$R=[r_{ij}]$ とすると，成分 r_{ij} は

$$r_{ij}=\begin{cases}
a_{m+j-i} & 1\leq j\leq n \text{ かつ } j\leq i\leq j+m\\
b_{j-i} & n+1\leq j\leq m+n \text{ かつ } j-n\leq i\leq j\\
0 & \text{その他}
\end{cases}$$

となる．複雑でどのような行列なのかわかりにくいと思われるので $m=3, n=5$ の場合を正確に書くと次のようになる．

$$\begin{bmatrix} a_3 & 0 & 0 & 0 & 0 & b_5 & 0 & 0 \\ a_2 & a_3 & 0 & 0 & 0 & b_4 & b_5 & 0 \\ a_1 & a_2 & a_3 & 0 & 0 & b_3 & b_4 & b_5 \\ a_0 & a_1 & a_2 & a_3 & 0 & b_2 & b_3 & b_4 \\ 0 & a_0 & a_1 & a_2 & a_3 & b_1 & b_2 & b_3 \\ 0 & 0 & a_0 & a_1 & a_2 & b_0 & b_1 & b_2 \\ 0 & 0 & 0 & a_0 & a_1 & 0 & b_0 & b_1 \\ 0 & 0 & 0 & 0 & a_0 & 0 & 0 & b_0 \end{bmatrix}$$

この $m+n$ 次行列の行列式を $\mathrm{Res}(f,g)$ と書いて f,g の**終結式** (resultant) という．この行列の $m+n$ 個の列ベクトルが線形独立であるための必要十分条件は行列式が 0 となることであるから，命題 3.2 により次の定理が得られる．

定理 3.3 多項式 f,g が共通因子を持つための必要十分条件は $\mathrm{Res}(f,g)=0$ となることである．

問 3.4 2 つの整数係数 2 次多項式 f,g の終結式 $\mathrm{Res}(f,g)$ の値を計算するプログラムを作成せよ．

4 判別式

n 次多項式

$$f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$$

の**判別式** D は，$f(x)=0$ の解を重複を込めて $\alpha_1,\ldots,\alpha_n$ とすると，

$$D=a_n^{2n-2}\prod_{1\le i<j\le n}(\alpha_i-\alpha_j)^2$$

で定義される．これは $\alpha_1,\ldots,\alpha_n$ についての対称式であることから $a_0,a_1,\ldots,a_n$ の多項式に書き直すことが出来る．定義から $D=0$ となるのは，ある $i<j$ について $\alpha_i=\alpha_j$ となることが必要十分条件となる

例 4.1 $n=2$, $f(x)=ax^2+bx+c$ とすると，$f(x)=0$ の解を α,β として $D=a^2(\alpha-\beta)^2=a^2(\alpha+\beta)^2-4a^2\alpha\beta$ となる．解と係数の関係 $\alpha+\beta=-b/a$, $\alpha\beta=c/a$ を使えば $D=b^2-4ac$ となる．

問 4.2 $n=3$, $f(x)=x^3-bx^2+cx-d$ について，$f(x)=0$ の解と係数の関係

$$\begin{aligned} b &= \alpha+\beta+\gamma \\ c &= \alpha\beta+\alpha\gamma+\beta\gamma \\ d &= \alpha\beta\gamma \end{aligned}$$

を用いて

$$D=b^2c^2-4c^3-4b^3d-27d^2+18bcd$$

となることを確かめよ．

この問題の計算は大変になる．D を求めるには次の式を順に b,c,d で記述する必要がある (第 7 節の対称多項式を参照)．

1. $\alpha^2+\beta^2+\gamma^2$
2. $\alpha^2\beta+\alpha^2\gamma+\alpha\beta^2+\alpha\gamma^2+\beta^2\gamma+\beta\gamma^2$
3. $\alpha^3+\beta^3+\gamma^3$
4. $\alpha^3\beta^3+\alpha^3\gamma^3+\beta^3\gamma^3$
5. $\alpha^4\beta^2+\alpha^4\gamma^2+\alpha^2\beta^4+\alpha^2\gamma^4+\beta^4\gamma^2+\beta^2\gamma^4$

次の定理は判別式の定義から明らかである．

定理 4.3 多項式 $f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$ の判別式 $D(a_0,a_1,\ldots,a_n)$ の $a_0,a_1,\ldots,a_n$ に数値を代入して値が 0 となるのは，$a_n=0$ かまたは方程式 $f(x)=0$ が重根を持つ場合である．

方程式 $f(x)=0$ が重根を持つことは，関数 $y=f(x)$ が x 軸に接することであるから，これは $f(x)$ と微分 $f'(x)$ が共通因子を持つことと同値である．これは終結式を用いて調べることができる．

問 4.4 $f(x)=ax^2+bx+c$ と $f(x)=x^3-bx^2+cx-d$ について終結式 $\mathrm{Res}(f(x),f'(x))$ を計算せよ．

5 単項式順序

n 変数多項式は有限個の単項式に係数をつけて加えた形の式であるが，n 変数の単項式は $x_1^{e_1}x_2^{e_2}\cdots x_{n-1}^{e_{n-1}}x_n^{e_n}$ と書けるので，これに $\mathbf{Z}_0^n$ の元 $(e_1,\ldots,e_n)$ が対応させることにより，単項式全体の集合から $\mathbf{Z}_0^n$ への全単射が得られる．逆写像は $\alpha=(e_1,\ldots,e_n)\in\mathbf{Z}_0^n$ に

$$x^\alpha=x_1^{e_1}x_2^{e_2}\cdots x_{n-1}^{e_{n-1}}x_n^{e_n}$$

を対応させる写像である．単項式 x^α の全次数 $\deg x^\alpha$ を $\deg x^\alpha = e_1 + \cdots + e_n$ で定義する．

1 変数の多項式を書く場合，次数の高い項から書くのが標準的であるが，2 変数以上の場合は書き方はいろいろあり得る．単に二つの多項式が等しいかどうかを調べる場合も，項の並び方が不統一では比較しにくい．そこで単項式全体の集合 $\{x^\alpha\,;\,\alpha \in \mathbf{Z}_0^n\}$ に全順序を入れておけば，多項式を記述する場合にこの順序により大きい単項式の項から順に並べることにより記述が一意的になる．この全順序が条件

$$x^\alpha > x^\beta \text{ ならば任意の } \gamma \in \mathbf{Z}_0^n \text{ について } x^\alpha x^\gamma > x^\beta x^\gamma \tag{6}$$

を満たすときこれを**単項式順序**という．なお，この条件から等号を含めた「$x^\alpha \geq x^\beta$ なら $x^\alpha x^\gamma \geq x^\beta x^\gamma$」の条件も得られる．

変数 $x_1, \ldots, x_n$ について条件 $x_1 > x_2 > \cdots > x_n > 1$ を満たす単項式順序として次のものがある．記号は Singular [1] に合わせている．

辞書式順序 (lex): $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\beta = (\beta_1, \ldots, \beta_n)$ とするとき

$$x^\alpha >_{\mathrm{lp}} x^\beta \Leftrightarrow \text{ ある } 1 \leq i \leq n \text{ について } \alpha_1 = \beta_1, \ldots, \alpha_{i-1} = \beta_{i-1} \text{ かつ } \alpha_i > \beta_i\,.$$

次数付き逆辞書式順序 (degrevlex): $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\beta = (\beta_1, \ldots, \beta_n)$ とするとき

$$\begin{aligned} &x^\alpha >_{\mathrm{dp}} x^\beta \\ &\quad\Leftrightarrow\ \deg x^\alpha > \deg x^\beta \text{ であるか，または } \deg x^\alpha = \deg x^\beta \text{ で，ある } 1 \leq i \leq n \\ &\qquad \text{について } \alpha_{i+1} = \beta_{i+1}, \ldots, \alpha_n = \beta_n \text{ かつ } \alpha_i < \beta_i\,. \end{aligned}$$

次数付き辞書式順序 (deglex): $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\beta = (\beta_1, \ldots, \beta_n)$ とするとき

$$\begin{aligned} &x^\alpha >_{\mathrm{Dp}} x^\beta \\ &\quad\Leftrightarrow\ \deg x^\alpha > \deg x^\beta \text{ であるか，または } \deg x^\alpha = \deg x^\beta \text{ で，ある } 1 \leq i \leq n \\ &\qquad \text{について } \alpha_1 = \beta_1, \ldots, \alpha_{i-1} = \beta_{i-1} \text{ かつ } \alpha_i > \beta_i\,. \end{aligned}$$

問 5.1 これらが単項式順序の条件 (6) を満たすことを確認せよ．

単項式が等しい場合を含めるため，それぞれ $\geq_{\mathrm{lp}}, \geq_{\mathrm{dp}}, \geq_{\mathrm{Dp}}$ の記号を使う場合もある．また，文献 [2] では $>_{\mathrm{lp}}$ は $>_{\mathrm{lex}}$ と記しており，$>_{\mathrm{dp}}$ は $>_{\mathrm{drlex}}$, $>_{\mathrm{Dp}}$ は $>_{\mathrm{dlex}}$ と記述している．

例 5.2 単項式 $x_1x_2^3x_3$, $x_1^2x_2x_3^2$, x_1^4 については

$$x_1^4 >_{\mathrm{lp}} x_1^2x_2x_3^2 >_{\mathrm{lp}} x_1x_2^3x_3$$

$$x_1x_2^3x_3 >_{\mathrm{dp}} x_1^2x_2x_3^2 >_{\mathrm{dp}} x_1^4$$

$$x_1^2x_2x_3^2 >_{\mathrm{Dp}} x_1x_2^3x_3 >_{\mathrm{Dp}} x_1^4$$

となる．辞書式順序では x_1 の指数が大きいことが第一の判定要素である．次数付き逆辞書式順序では x_1^4 の全次数が 4 で最小となり，$x_1x_2^3x_3$ と $x_1^2x_2x_3^2$ では全次数は 5 で同じであるから x_3 の次数を比べて小さい方の $x_1x_2^3x_3$ を順序を上とする．次数付き辞書式順序では x_1 の次数を比べて大きい方の $x_1^2x_2x_3^2$ を順序を上とする．

問 5.3 x_1, x_2, x_3 の全次数 4 の単項式全体を各順序について大きい順に並べよ．

全順序ではないが，$\mathbf{Z}_0^n$ の元 $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\beta = (\beta_1, \ldots, \beta_n)$ に対して

$$\alpha \geq_{\mathrm{nat}} \beta \Leftrightarrow \alpha_i \geq \beta_i \ (i = 1, \ldots, n)$$

で定義される $\mathbf{Z}_0^n$ の順序を**自然順序**という．単項式についても $\alpha \geq_{\mathrm{nat}} \beta$ の場合に $x^\alpha \geq_{\mathrm{nat}} x^\beta$ と書いて自然順序という．単項式順序の場合は $x^\alpha \geq x^\beta$ でも $x^{\alpha-\beta}$ が $K[x_1, \ldots, x_n]$ の単項式とは限らないが，$x^\alpha \geq_{\mathrm{nat}} x^\beta$ の場合は $x^{\alpha-\beta} \in K[x_1, \ldots, x_n]$ となる．

補題 5.4 (ディクソン) $\mathbf{Z}_0^n$ の任意の部分集合 S について S の自然順序での極小元は有限個である．

証明　背理法で示す．極小元が無限個あるとすると，S の相異なる極小元からなる無限列 $\{\alpha^{(i)}\}$ が存在する．この無限列から，各 $1 \leq j \leq n$ に対して非負整数の数列 $\{\alpha_j^{(i)}\}$ が得られる．この数列が無限個の非負整数の値を取るとすると，どの項にも後に必ずそれより大きな項があるので部分列を取って i について単調増加にできる．一方，もしこの数列が有限個の値しか取らなければすべて同じ値となる部分列が取れる．これらのことは $\{\alpha^{(i)}\}$ の任意の部分列についても行えるので，$j = 1, 2, \ldots, n$ の順にこのような部分列を取り，得られた部分列を $\{\beta^{(i)}\}$ とする．このとき，i についての数列 $\{\beta_j^{(i)}\}$ はすべての $j = 1, 2, \ldots, n$ に対して広義単調増加であるから $\beta^{(2)} \geq_{\mathrm{nat}} \beta^{(1)}$ となる．$\beta^{(2)} \neq \beta^{(1)} \in S$ であるから $\beta^{(2)}$ が S の極小元であることに矛盾する．　証明終わり

$\mathbf{Z}_0^n$ の部分集合 S が条件「$\alpha \in S, \beta \in \mathbf{Z}_0^n$ ならば $\alpha + \beta \in S$」を満たすとき S を $\mathbf{Z}_0^n$ のイデアルという．半群 $\mathbf{Z}_0^n$ のイデアルという意味なので，本書では可換環のイデアルと区別するため半群イデアルと呼ぶ．

系 5.5 $\mathbf{Z}_0^n$ の任意の半群イデアル S に対して有限個の元 $\alpha_1, \ldots, \alpha_s \in S$ が存在して

$$S = (\alpha_1 + \mathbf{Z}_0^n) \cup \cdots \cup (\alpha_s + \mathbf{Z}_0^n)$$

となる．ここで $\alpha_i + \mathbf{Z}_0^n = \{\alpha_i + \beta \,;\, \beta \in \mathbf{Z}_0^n\}$ である．$\{\alpha_1, \ldots, \alpha_s\}$ は自然順序での S の極小元全体に取れる，

証明　$\{\alpha_1, \ldots, \alpha_s\}$ を S の自然順序での極小元全体とする．補題 5.4 によりこれは有限集合である．S が半群イデアルで $\alpha_i \in S$ であるから $\alpha_i + \mathbf{Z}_0^n \subset S$ となり，右辺は S に含まれる．一方 S の任意の元 α に対して $\alpha \geq_{\mathrm{nat}} \beta$ となる $\mathbf{Z}_0^n$ の元 β は有限個であるから，そのような S の元も有限個である．その中の極小元は S でも極小なので，その 1 つを α_i とすれば $\alpha \in \alpha_i + \mathbf{Z}_0^n$ となり α が右辺に含まれることもわかる．　証明終わり

問 5.6 正の整数 d に対して $S = \{(a, b) \in \mathbf{Z}_0^2 \,;\, ab \geq d\}$ が $\mathbf{Z}_0^2$ の半群イデアルとなることを示せ．また $d = 5$ の場合と $d = 25$ の場合に S の極小元をすべて求めよ．

定理 5.7 $>$ を $x_1, \ldots, x_n > 1$ となる単項式順序とすると，$x^\alpha \geq_{\mathrm{nat}} x^\beta$ なら $x^\alpha \geq x^\beta$ である．また，単項式全体はこの順序について整列集合である．

証明　$x^\alpha \geq_{\mathrm{nat}} x^\beta$ とする．$\alpha - \beta = (c_1, \ldots, c_n)$ とすると，$x_1, \ldots, x_n > 1$ の仮定から $x_1^{c_1} \geq 1, \ldots, x_n^{c_n} \geq 1$ となるので (6) を繰り返し適用して

$$x^{\alpha-\beta} = x_1^{c_1} \cdots x_n^{c_n} \geq x_1^{c_1} \cdots x_{n-1}^{c_{n-1}} \geq \cdots \geq x_1^{c_1} \geq 1$$

となる．両辺に x^β をかけて $x^\alpha \geq x^\beta$ がわかる．整列集合であることは，S を単項式からなる空でない集合としたとき S に最小元があることを示せばよい．$\{x^{\alpha_1}, \ldots, x^{\alpha_s}\}$ を自然順序での S の極小元全体とする．$>$ は全順序有限集合なのでこの中に最小元 x^{α_i} がある．このとき x^{α_i} は S でも最小である．実際，任意の $x^\beta \in S$ について $x^\beta \geq_{\mathrm{nat}} x^{\alpha_j}$ となる j があり，$x^\beta \geq x^{\alpha_j} \geq x^{\alpha_i}$ となる．　証明終わり

6　グレブナー基底

可換環 R の部分加群 I が条件「$u \in I, r \in R$ ならば $ru \in I$」を満たすとき，I を R のイデアルという．

R の任意の部分集合 S について

$$I = \{r_1 u_1 + \cdots + r_l u_l \,;\, u_1, \ldots, u_l \in S, r_1, \ldots, r_l \in R\}$$

は R のイデアルである．また，この I に対して S を I の生成系と言う．任意のイデアルに対して有限個の元からなる生成系が存在する環をネーター環と言う．体上の多項式環など，この講義に出てくる可換環はすべてネーター環である．多項式環のネーター性は定理 6.5 で証明される．

$x_1 > x_2 > \cdots > x_n > 1$ を満たす単項式順序 $>$ を固定する．$x^\alpha > x^\beta$ または $x^\alpha = x^\beta$ であることを $x^\alpha \geq x^\beta$ と書く．これまで同様に $\alpha = (a_1, \ldots, a_n) \in \mathbf{Z}_0^n$ に対して $x^\alpha = x_1^{a_1} \cdots x_n^{a_n}$ であり，$x^\alpha > x^\beta$ のときに x^α の順序が x^β より「大きい」,「高い」,「前にある」のように言う．仮定から 1 が一番順序の低い単項式である．

0 でない多項式 $f \in K[x_1, \ldots x_n]$ について，順序の一番大きい単項式の項 cx^α を先頭項と言い $\mathrm{LT}(f)$ と書く．このほか，次の記号を用いる．

$$\mathrm{LM}(f) = x^\alpha\,,\ \ \mathrm{LE}(f) = \alpha\,,\ \ \mathrm{LC}(f) = c\,,\ \ \mathrm{tail}(f) = f - \mathrm{LT}(f)$$

なお，英語の対応は LM : leading term, LM : leading monomial, LC : leading coefficient である．

次は容易にわかる．

補題 6.1 $f \in K[x_1, \dots x_n] \setminus \{0\}$, $\beta \in \mathbf{Z}_0^n$ に対して

$$\mathrm{LT}(x^\beta f) = x^\beta \,\mathrm{LT}(f)\ ,\ \ \mathrm{LM}(x^\beta f) = x^\beta \,\mathrm{LM}(f)\ ,\ \ \mathrm{LE}(x^\beta f) = \mathrm{LE}(f) + \beta$$

となる．さらに $f, g \in K[x_1, \dots x_n] \setminus \{0\}$ に対して

$$\mathrm{LT}(fg) = \mathrm{LT}(f)\,\mathrm{LT}(g)\ ,\ \ \mathrm{LM}(fg) = \mathrm{LM}(f)\,\mathrm{LM}(g)\ ,\ \ \mathrm{LE}(fg) = \mathrm{LE}(f) + \mathrm{LE}(g)$$

が成り立つ．

多項式 f, g が $\mathrm{LM}(f) \geq \mathrm{LM}(g)$ を満たせば $u = \mathrm{LT}(f)/\mathrm{LT}(g)$ は K の 0 でない元を係数とする単項式である．$\mathrm{LT}(f) = \mathrm{LT}(ug)$ であるから $\mathrm{LM}(f - ug) < \mathrm{LM}(f)$ となる．これが単項式順序を入れた多項式環での，多項式の先頭項を消去して下げる基本操作である．

多項式環 $K[x_1, \dots x_n]$ のイデアル I について $\{\mathrm{LM}(f)\,;\, f \in I \setminus \{0\}\}$ で生成されるイデアルを I の先頭項イデアルと言い $L(I)$ と書く．

$L(I)$ については $\mathrm{LE}(I) = \{\mathrm{LE}(f)\,;\, f \in I \setminus \{0\}\}$ を考えるとわかりやすい．

補題 6.2 I を $K[x_1, \dots x_n]$ のイデアルとすると $\mathrm{LE}(I)$ は $\mathbf{Z}_0^n$ の半群イデアルで，先頭項イデアル $L(I)$ は $\{x^\alpha\,;\, \alpha \in \mathrm{LE}(I)\}$ を基底とする $K[x_1, \dots x_n]$ の部分ベクトル空間 V に一致する．

証明　任意の $\alpha \in \mathrm{LE}(I)$ に対して $f \in I$ があって $x^\alpha = \mathrm{LM}(f)$ であるから $x^\alpha \in L(I)$ である．よってこれらで生成されるベクトル空間 V はイデアル $L(I)$ に含まれる．また，上記の α と任意の $\beta \in \mathbf{Z}_0^n$ に対して，$x^\beta f \in I$ より $\alpha + \beta = \mathrm{LE}(x^\beta f) \in \mathrm{LE}(I)$ であるから $\mathrm{LE}(I)$ は半群イデアルである．$L(I)$ の生成系 $\{\mathrm{LM}(f)\,;\, f \in I \setminus \{0\}\}$ が V に含まれることは明らかなので，V がイデアルであることを示せば $L(I) = V$ がわかる．任意の元 $f \in V$ と $g \in K[x_1, \dots x_n]$ に対して，f の各単項式はある x^α $(\alpha \in \mathrm{LE}(I))$ の定数倍で g の各単項式は x^β $(\beta \in \mathbf{Z}_0^n)$ の定数倍であるから，fg は $x^{\alpha+\beta} \in V$ の定数倍の和であり，$\alpha + \beta \in \mathrm{LE}(I)$ なので V に含まれる．したがって V は $K[x_1, \dots x_n]$ のイデアルであり，$L(I) = V$ となる．　証明終わり

$K[x_1, \dots x_n] \setminus \{0\}$ の有限部分集合 $G = \{g_1, \dots, g_s\}$ に対して，$\{\mathrm{LM}(g_1), \dots, \mathrm{LM}(g_s)\}$ で生成されるイデアルを $L(G)$ と書く．G で生成されるイデアルを I とすると明らかに $L(G) \subset L(I)$ であるが，等号は一般には成り立たない．例えば $K[x]$ について $G = \{x, x - 1\}$ とすれば，$I = K[x]$ であるから $L(I) = K[x]$ である一方，$\mathrm{LM}(x) = \mathrm{LM}(x - 1) = x$ より $L(G) = xK[x]$ である．

イデアル I の 0 を含まない有限部分集合 $G = \{g_1, \dots, g_s\}$ について，$L(G) = L(I)$ であるとき G を I の**グレブナー基底**と言う．$\mathrm{LE}(I)$ は $\mathbf{Z}_0^n$ の半群イデアルで $\{\alpha_1, \dots, \alpha_s\}$ をその自然順序での極小元全体とすれば，各 α_i に対して $g_i \in I$ が存在して $\mathrm{LE}(g_i) = \alpha_i$ となる．このとき系 5.5 により $\{\mathrm{LM}(g_1), \dots, \mathrm{LM}(g_s)\}$ は $L(I)$ を生成するので $G = \{g_1, \dots, g_s\}$ は I のグレブナー基底となる．すなわちグレブナー基底は必ず存在する．

補題 6.3 $G = \{g_1, \ldots, g_s\}$ を $K[x_1, \ldots x_n] \setminus \{0\}$ の有限部分集合とする．$K[x_1, \ldots x_n]$ の任意の元 f に対して，$f_1, \ldots, f_s \in K[x_1, \ldots x_n]$ で

$$f_i \neq 0 \text{ ならば } \mathrm{LM}(f_i g_i) \leq \mathrm{LM}(f) \quad (i = 1, \ldots, s) \tag{7}$$

であって $f - (f_1 g_1 + \cdots + f_s g_s)$ の単項式がいずれも $L(G)$ に含まれないものが存在する．

証明　$f = 0$ なら $f_1, \ldots, f_s = 0$ でよいので $f \neq 0$ とする．この補題の主張が成り立たない f が存在したとして，f として $\mathrm{LM}(f)$ が最小となるものをとる．$\mathrm{LM}(f) \in L(G)$ の場合は，ある i について $\mathrm{LE}(f) \in (\mathrm{LE}(g_i) + \mathbf{Z}_0^n)$ であるから，$u = \mathrm{LT}(f)/\mathrm{LT}(g_i)$ とおけば $\mathrm{LM}(f - ug_i) < \mathrm{LM}(f)$ となる．$\mathrm{LM}(f)$ の最小性から $f - ug_i$ については補題の条件を満たす $f'_1, \ldots, f'_s$ が存在する．$f_i = f'_i + u$ とし，$j \neq i$ については $f_j = f'_j$ と置けば，$\mathrm{LM}(f'_i g_i) < \mathrm{LM}(f)$, $\mathrm{LM}(ug_i) = \mathrm{LM}(f)$ より $\mathrm{LM}(f_i g_i) = \mathrm{LM}(f)$ で (7) が確認されるとともに

$$f - (f_1 g_1 + \cdots + f_s g_s) = (f - ug_i) - (f'_1 g_1 + \cdots + f'_s g_s)$$

となり，f に対する補題の条件を満たす．しかし f についての仮定からこれはあり得ない．$\mathrm{LM}(f) \notin L(G)$ とする．$\mathrm{LM}(f - \mathrm{LT}(f)) < \mathrm{LM}(f)$ であるから $\mathrm{LM}(f)$ の最小性により，$f - \mathrm{LT}(f)$ について補題の条件を満たす $f_1, \ldots, f_s$ が存在する．

$$f - (f_1 g_1 + \cdots + f_s g_s) = (f - \mathrm{LT}(f)) - (f_1 g_1 + \cdots + f_s g_s) + \mathrm{LT}(f)$$

であるから，これらは f に対しても条件を満たし，これも仮定に矛盾する．　　証明終わり

例 6.4 この補題の $f_1, \ldots, f_s$ の取り方も $f_1 g_1 + \cdots + f_s g_s$ も一意的ではない．$n = 1$ で $G = \{x, x - 1\}$ とすると $L(G) = xK[x]$ であり，$f = x$ について $f_1 = 1$, $f_2 = 0$ とすれば $f - (f_1 g_1 + f_2 g_2) = 0$ であり，また $f_1 = 0$, $f_2 = 1$ とすれば $f - (f_1 g_1 + f_2 g_2) = 1$ となり，どちらを用いてもよい．

次の定理は多項式環についてのヒルベルトの基底定理を含む．

定理 6.5 $G = \{g_1, \ldots, g_s\}$ をイデアル I のグレブナー基底とすると，I は G で生成される．グレブナー基底は存在するので $K[x_1, \ldots x_n]$ の任意のイデアルは有限生成である．

証明　$L(G)$ を G で生成されるイデアルとする．$G \subset I$ であるから $L(G) \subset I$ である．元 $f \in I \setminus L(G)$ があったとする．補題 6.3 により $g \in L(G)$ があって $\mathrm{LM}(f - g)$ が $L(I)$ に含まれない．$f - g \in I$ であるからこれは $L(I)$ の定義に矛盾する．　　証明終わり

計算機に $\mathrm{LT}(f)$ や $\mathrm{LM}(f)$ を求める手続きが用意されていたとする．$G = \{g_1, \ldots, g_s\}$ の順序を決めることにより，補題 6.3 の $g = f_1 g_1 + \cdots + f_s g_s$ を求めるアルゴリズムが次のように記述できる．

INPUT: f

OUTPUT: g, $h = f - g$
初期設定： $g = 0$, $h = 0$
LOOP:
(1) $\mathrm{LM}(f) \notin L(G)$ であれば $f := f - \mathrm{LT}(f)$, $h := h + \mathrm{LT}(f)$ として，$f = 0$ なら LOOP を終わり，$f \neq 0$ なら LOOP を続ける．
(2) $\mathrm{LM}(f) \in L(G)$ であれば $i = 1, \ldots, s$ の順に $\mathrm{LE}(f) \geq_{\mathrm{nat}} \mathrm{LE}(g_i)$ となる i を探し，最初の i について $f := f - (\mathrm{LT}(f)/\mathrm{LT}(g_i))g_i$, $g := g + (\mathrm{LT}(f)/\mathrm{LT}(g_i))g_i$ として，$f = 0$ なら LOOP を終了し，$f \neq 0$ なら LOOP を続ける．

アルゴリズムの途中で f, g, h は変化するが $f + g + h$ は不変で常に f の初期値に等しい．(1) の操作から h の単項式はすべて $L(G)$ の外にある．(2) の操作から g は常にイデアル I の元である．LOOP により毎回 $\mathrm{LM}(f)$ は小さくなるので，多項式順序の整列性から LOOP は必ず終了する．終了したときに $f = 0$ であるから $g + h$ は f の初期値に等しく補題 6.3 の条件を満たす．(2) の操作を記録しておけば $f_1, \ldots, f_s$ も同時に求まる．また，補題 6.3 の条件を「$\mathrm{LM}(f - g)$ が $L(G)$ に含まれない」に弱めた場合，(1) となる f が出た時点で終了して，そのときの g を取ればよい．下記の補題 6.6 を含めて，この弱めた条件で求めた g で十分な場合が多い．

このアルゴリズムにより得られた $h = f - g$ を $\overline{f}^G$ と書く．$g = f_1 g_1 + \cdots + f_s g_s \in I$ であるから

$$f \equiv \overline{f}^G \pmod{I} \tag{8}$$

が成り立つ．1 変数で $G = \{g\}$ の場合は，f を g で割ることにより多項式 p, q $(\deg q < \deg g)$ があって $f = pg + q$ と書ける．この場合 $\overline{f}^G = q$ である．ここで構成した $\overline{f}^G$ はこの 1 変数の場合の割り算を一般化して，多変数の多項式を与えられた多項式の列 G で割れる部分を可能な限り割って出した余りと考えることが出来る．ただし $\overline{f}^G$ は G の元の番号付けによる順序にも依存して定義されていることに注意が必要である．

補題 6.6 $G = \{g_1, \ldots, g_s\}$ がイデアル I のグレブナー基底とすると，任意の $f \in K[x_1, \ldots, x_n]$ について，$\overline{f}^G = 0$ であることは $f \in I$ と同値である．

証明　$\overline{f}^G = 0$ であれば (8) により $f \in I$ となる．もし $\overline{f}^G \neq 0$ であれば補題 6.3 により $\mathrm{LM}(\overline{f}^G) \notin L(G)$ であるが，G がグレブナー基底であるから $L(G) = L(I)$ で $\overline{f}^G \notin I$ となる．よって (8) により $f \notin I$ となる．　証明終わり

計算機で多項式環を扱う場合，イデアルは生成系，つまり有限個の生成元を並べたリストで与えられる．ここでリストというのは $G = \{g_1, \ldots, g_s\}$ のように元の順序を決めた有限集合を意味する．補題 6.6 はイデアル I がグレブナー基底 G で与えられた場合は，$\overline{f}^G$ の計算方法を計算機にプログラムとして実装すれば，任意の多項式 f について $f \in I$ が計算機により判定可能となることを示している．

B が可換環 A の部分環で I が A のイデアルとすると $I \cap B$ は B のイデアルとなる．$A = K[x_1, \ldots, x_n]$ で，ある $1 \leq s < n$ について $B = K[x_{s+1}, \ldots, x_n]$ の場合，$I \cap B$ を求める

ことは，$G = \{g_1, \ldots, g_l\}$ が I の生成系とすれば，方程式系

$$g_1 = \cdots = g_l = 0$$

から変数 $x_1, \ldots, x_s$ を消去することと同じ意味になる．

単項式順序 $>$ は $\mathrm{LM}(f) \in K[x_{s+1}, \ldots, x_n]$ ならば $f \in K[x_{s+1}, \ldots, x_n]$ を満たす場合 $x_1, \ldots, x_s$ について**消去順序**であるという．なお $K[x_{s+1}, \ldots, x_n]$ は $K[x_1, \ldots, x_n]$ の部分環であるが，任意の単項式順序は $K[x_{s+1}, \ldots, x_n]$ の単項式全体に制限してもまた単項式順序となる．$f \in K[x_{s+1}, \ldots, x_n]$ に対する $\mathrm{LT}(f)$ や $\mathrm{LM}(f)$ は $K[x_1, \ldots, x_n]$ で考えても，$K[x_{s+1}, \ldots, x_n]$ に制限された順序で考えても同じである．

例 6.7 (1) 辞書式順序 $>_{\mathrm{lp}}$ では $x_1, \ldots, x_s$ の順序は $K[x_{s+1}, \ldots, x_n]$ のどの単項式よりも上なので，$\mathrm{LM}(f) \in K[x_{s+1}, \ldots, x_n]$ であれば f に変数 $x_1, \ldots, x_s$ は現れない．これは $f \in K[x_{s+1}, \ldots, x_n]$ を意味する．したがって任意の $1 \leq s < n$ について $>_{\mathrm{lp}}$ は消去順序となる．

(2) $>$ を任意の単項式順序とする．順序 $>'$ を $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\beta = (\beta_1, \ldots, \beta_n)$ に対して

$$\begin{aligned} x^\alpha >' x^\beta \quad \Leftrightarrow \quad & \alpha_1 + \cdots + \alpha_s > \beta_1 + \cdots + \beta_s \text{ であるか，または} \\ & \alpha_1 + \cdots + \alpha_s = \beta_1 + \cdots + \beta_s \text{ かつ } x^\alpha > x^\beta \end{aligned}$$

で定義すると，これは単項式順序であることが容易に確認できて，しかも $x_1, \ldots, x_s$ についての消去順序となる．実際，判定条件の指数の和が s までであるから変数 $x_1, \ldots, x_s$ の現れる単項式の順序は $K[x_{s+1}, \ldots, x_n]$ のどの単項式より順序が上となる．したがって $>'$ は (1) と同様に消去順序となる．

定理 6.8 $>$ が $x_1, \ldots, x_s$ について消去順序であるとする．$G = \{g_1, \ldots, g_l\}$ がイデアル I のグレブナー基底で，

$$\mathrm{LM}(g_1), \ldots, \mathrm{LM}(g_t) \notin K[x_{s+1}, \ldots, x_n]\ , \ \ \mathrm{LM}(g_{t+1}), \ldots, \mathrm{LM}(g_l) \in K[x_{s+1}, \ldots, x_n]$$

であれば $G' = \{g_{t+1}, \ldots, g_l\}$ は $I \cap K[x_{s+1}, \ldots, x_n]$ のグレブナー基底となる．

証明　$>$ が消去順序であるから，条件より $g_{t+1}, \ldots, g_l \in K[x_{s+1}, \ldots, x_n]$ がわかる．任意の 0 でない $f \in I \cap K[x_{s+1}, \ldots, x_n]$ について $x^\alpha = \mathrm{LM}(f)$ とする．このとき G が I のグレブナー基底であることから $x^\alpha \in L(I) = L(G)$ となる．したがって，ある g_i について $x^\alpha \geq_{\mathrm{nat}} \mathrm{LM}(g_i)$ となる．x^α は $K[x_{s+1}, \ldots, x_n]$ に含まれる単項式なので $x_1, \ldots, x_s$ の指数は 0 である．したがって，この自然順序での不等式から $\mathrm{LM}(g_i)$ も同様で仮定から $g_i \in G'$ がわかる．よって $K[x_{s+1}, \ldots, x_n]$ の中で考えて $x^\alpha \in L(G')$ となり $L(G') = L(I \cap K[x_{s+1}, \ldots, x_n])$ がわかる．

証明終わり

7　対称多項式

この節では整数係数の多項式を考える．S_n を n 次対称群とする．ただし置換 $\sigma, \tau \in S_n$ の積は永尾氏の教科書「代数学」p.8 と同じく $(\sigma\tau)(i) = \tau(\sigma(i))$ で定義されるものとする．

n 変数の多項式 $f(x_1, \ldots, x_n) \in \mathbf{Z}[x_1, \ldots, x_n]$ は任意の $\sigma \in S_n$ に対して $f(x_{\sigma(1)}, \ldots, x_{\sigma(n)}) = f(x_1, \ldots, x_n)$ を満たすとき対称多項式または対称式と言う．$f^\sigma(x_1, \ldots, x_n) = f(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$ と書けば，置換の積の定義により $f^{\sigma\tau} = (f^\sigma)^\tau$ となる．

例 7.1 2 変数多項式の場合，多項式 $f(x, y)$ は $f(y, x) = f(x, y)$ となるとき対称多項式である．例えば

$$x + y\ ,\quad xy\ ,\quad x^2 + y^2\ ,\quad 3x^3 - 2x^2y - 2xy^2 + 3y^3 + 1$$

は対称式である．一方 $f(x, y) = x^2 - y^2$ は $f(y, x) = y^2 - x^2 = -x^2 + y^2$ となるので対称式ではない．

命題 7.2 2 変数の対称多項式 $f(x, y)$ は

$$f(x, y) = \sum_{i \geq 1, j \geq 0} a_{ij}(x^i + y^i)x^j y^j + \sum_{j \geq 0} b_j x^j y^j \tag{9}$$

の形に一意的に書ける．また，この右辺の形の式は常に対称式である．

証明　この形の式が対称式であるのは明らかである．

$f(x, y)$ を対称式として，(9) の形に書けることを項の数についての数学的帰納法で示す．項の数が 0 であれば $f(x, y) = 0$ であるからすべて $a_{ij} = b_j = 0$ でよい．$f(x, y)$ が項 $ax^p y^q$ を含むとすると，$p \neq q$ なら対称性から $f(x, y)$ は x と y を入れ替えた $ax^q y^p$ を項として含む．$ax^p y^q + ax^q y^p$ は $p > q$ なら $a(x^{p-q} + y^{p-q})x^q y^q$, $p < q$ なら $a(x^{q-p} + y^{q-p})x^p y^p$ となり，いずれにしても (9) の形の対称式で $f(x, y) - (ax^p y^q + ax^q y^p)$ は $f(x, y)$ より項の数が 2 つ少ない対称式となる．帰納法の仮定からこの式は (9) の形に書ける．$p = q$ なら $f(x, y) - ax^p y^p$ が (9) の形になる．したがって $f(x, y)$ もこの形に書ける．a_{ij} は $x^{i+j}y^j$ の係数，b_j は $x^j y^j$ の係数であるから一意的である．　証明終わり

定理 7.5 で一般変数の場合として示すように 2 変数の対称多項式は $x + y$ と xy の多項式として書ける．例えば $x^d + y^d$ $(d = 2, 3, 4)$ については

$$\begin{aligned} x^2 + y^2 &= (x + y)^2 - 2xy\ , \\ x^3 + y^3 &= (x + y)^3 - 3(x + y)xy\ , \\ x^4 + y^4 &= (x + y)^4 - 4(x + y)^2 xy + 2x^2 y^2 \end{aligned}$$

となることが容易に確かめられる．

問 7.3 $5 \leq d \leq 12$ について $x^d + y^d$ を $x + y$ と xy の式で表せ．

上の問を低い d から順に計算すれば効率の良い計算法が見つかるはずである．係数の桁数も増えてくるので，高い d については計算機を使うべき問題とわかる．

問 7.4 一般の $d > 0$ について $x^d + y^d$ を $x + y$ と xy の式で表す式を求めるプログラムを C で作成せよ．ただし，C で扱える整数は絶対値 10 億ぐらいまでなので，d については限度がある．

この問のプログラムの例を以下で記述する．
$(x+y)^i(xy)^j$ は斉次 $i+2j$ 次式であるから，斉次 d 次式の $x^d + y^d$ は

$$(x+y)^d\ ,\ \ (x+y)^{d-2}(xy)\ ,\ \ (x+y)^{d-4}(xy)^2\ , \ldots$$

の 1 次結合で書けるはずである．なお，最後の項は $d = 2e$ なら $(xy)^e$, $d = 2e+1$ なら $(x+y)(xy)^e$ である．

$$x^d + y^d = \sum_{j=0}^{e} a(d,j)(x+y)^{d-2j}(xy)^j \tag{10}$$

と置き，係数 $a(d,j)$ を求めたい．$d = 0, 1$ については

$$a(0,0) = 2\ ,\ \ a(1,0) = 1$$

である．一般には $d \geq 2$ として，等式

$$x^d + y^d = (x+y)(x^{d-1} + y^{d-1}) - (xy)(x^{d-2} + y^{d-2}) \tag{11}$$

が成り立つので，これを使って $a(d,j)$ について漸化式を求める．$d = 2e$ なら (11) に (10) の右辺を代入して

$$\begin{aligned}
&\sum_{j=0}^{e} a(d,j)(x+y)^{d-2j}(xy)^j \\
&= \sum_{j=0}^{e-1} a(d-1,j)(x+y)^{d-2j}(xy)^j - \sum_{j=0}^{e-1} a(d-2,j)(x+y)^{d-2-2j}(xy)^{j+1} \\
&= \sum_{j=0}^{e-1} a(d-1,j)(x+y)^{d-2j}(xy)^j - \sum_{j=1}^{e} a(d-2,j-1)(x+y)^{d-2j}(xy)^j \\
&= \sum_{j=0}^{e} \{a(d-1,j) - a(d-2,j-1)\}(x+y)^{d-2j}(xy)^j
\end{aligned}$$

となる．これから漸化式

$$a(d,j) = a(d-1,j) - a(d-2,j-1)$$

が得られる．$d = 2e+1$ の場合も同様に同じ漸化式が得られる．ただし，整数 j が $0 \leq j \leq d/2$ でなければ $a(d,j) = 0$ とする．

```
//
// 計算機数学 B プログラム 3
// sym2tex.c
// n = 2, 3,..., 39 について対称式 x^n+y^n を x+y, xy の多項式に表す
// TeX ファイルとして出力する

#include <stdio.h>

#define N 40
#define M 20

FILE *fp;

int main()
{
    int a[N][M];
    int i, j;
    fp = fopen("sym2.tex","w");
    fputs("\\documentclass[12pt]{jarticle}\n\n\\begin{document}\n\n", fp);
    for (i=0; i<N; i++) {
        for (j=0; j<M; j++) a[i][j] = 0;
    }
    a[0][0] = 2;
    a[1][0] = 1;
    for (i=2; i<N; i++) {
        a[i][0] = a[i-1][0];
        fprintf(fp, "$x^{%d}+y^{%d}=%d(x+y)^{%d}", i, i, a[i][0], i);
        for (j=1; j<=i/2; j++) {
            a[i][j] = a[i-1][j]-a[i-2][j-1];
            if (a[i][j]>0)
              fprintf(fp, "+%d(x+y)^{%d}(xy)^{%d}",a[i][j],i-2*j,j);
            else if (a[i][j]<0)
              fprintf(fp,"-%d(x+y)^{%d}(xy)^{%d}",-a[i][j],i-2*j,j);
        }
        fputs("$\n\n\\vspace{2mm}\n", fp);
    }
    fputs("\\end{document}", fp);
    fclose(fp);
    return 0;
```

}

$f(x_1,\ldots,x_n)$ を対称多項式とする．対称多項式の定義から $f(x_1,\ldots,x_n)$ が項 $ax_1^{e_1}\cdots x_n^{e_n}$ を持てば，任意の $\sigma\in S_n$ に対して

$$x_{\sigma(1)}^{e_1}\cdots x_{\sigma(n)}^{e_n}$$

の係数も a となる．

$$H(e_1,\ldots,e_n)=\{\sigma\in S_n\,;\,e_{\sigma(i)}=e_i,i=1,\ldots,n\}$$

と置けば，これは S_n の部分群で

$$x_{\sigma(1)}^{e_1}\cdots x_{\sigma(n)}^{e_n}=x_{\tau(1)}^{e_1}\cdots x_{\tau(n)}^{e_n}$$

は $\tau\sigma^{-1}\in H(e_1,\ldots,e_n)$ が必要十分条件となる．すなわち S_n の部分群 $H(e_1,\ldots,e_n)$ の同じ右剰余類に含まれる元について同じ単項式となる．したがって $\sigma\in S_n$ についての和

$$\sum_{\sigma\in S_n}x_{\sigma(1)}^{e_1}\cdots x_{\sigma(n)}^{e_n}$$

の各項の係数はすべて右剰余類の元の個数であるこの部分群の位数 $|H(e_1,\ldots,e_n)|$ に等しい．

単項式 $x_1^{e_1}\cdots x_n^{e_n}$ に対して指数 $e_1,\ldots,e_n$ を大きい順に並べ直したものを $e_1',\ldots,e_n'$ とし，$(e_1',\ldots,e_n')$ をこの単項式の型と呼ぶ．単項式 $x_1^{e_1}\cdots x_n^{e_n}$ と $x_1^{d_1}\cdots x_n^{d_n}$ の型が等しいことは，ある $\sigma\in S_n$ について $d_i=e_{\sigma(i)}$ $(i=1,\ldots,n)$ となることであるから，対称式は型の等しい単項式の係数がすべて等しい多項式として特徴づけられることがわかる．

$e_1\geq e_2\geq\cdots\geq e_n\geq 0$ を満たす整数列 $e_1,\ldots,e_n$ に対して

$$s(e_1,\ldots,e_n)=\frac{1}{|H(e_1,\ldots,e_n)|}\sum_{\sigma\in S_n}x_{\sigma(1)}^{e_1}\cdots x_{\sigma(n)}^{e_n}$$

と置く．前に注意したことから，これは係数がすべて 1 の対称多項式となる．

$i=1,\ldots,n$ に対して基本対称式 $s_1,\ldots,s_n$ を

$$\begin{aligned}
s_1 &= s(1,0,0,0,\ldots,0)\\
s_2 &= s(1,1,0,0,\ldots,0)\\
s_3 &= s(1,1,1,0,\ldots,0)\\
&\vdots\\
s_n &= s(1,1,1,1,\ldots,1)
\end{aligned}$$

で定義する．

定理 7.5 $f(x_1,\ldots,x_n)\in\mathbf{Z}[x_1,\ldots,x_n]$ が対称多項式となるための必要十分条件は，基本対称式 $s_1,\ldots,s_n$ の整数係数多項式となることである．

証明　$s_1, \ldots, s_n$ はすべて対称多項式であるから，これらの整数係数多項式も対称多項式となる．

対称多項式がすべて $s_1, \ldots, s_n$ の整数係数多項式となることを示す．対称多項式は $s(e_1, \ldots, e_n)$ の整数倍の和となるので，任意の $e_1 \geq e_2 \geq \cdots \geq e_n \geq 0$ に対して $s(e_1, \ldots, e_n)$ が $s_1, \ldots, s_n$ の整数係数多項式となること示せば良い．$d = e_1$ についての数学的帰納法で示す．

$d = 1$ の場合はある $1 \leq i \leq n$ について $s(e_1, \ldots, e_n) = s_i$ となるので正しい．$d > 1$ として $e_1 < d$ の場合は成り立つと仮定する．

$e_1 = d$ とする．もし $e_1 = \cdots = e_n$ であれば $s(e_1, \ldots, e_n) = s_n^d$ であるから正しい．$e_1 > e_n$ とする．このとき，ある $1 \leq p < n$ について $e_1 = \cdots = e_p > e_{p+1}$ となる．$s(e_1, \ldots, e_n)$ が $s_1, \ldots, s_n$ の整数係数多項式となることを，$d = e_1$ の仮定を置いて p についての数学的帰納法で示す．つまり，二重の数学的帰納法である．

最初の帰納法の仮定から $s(d-1, \ldots, d-1, e_{p+1}, \ldots, e_n)$ は $s_1, \ldots, s_n$ の整数係数多項式となる．

$$s(d, \ldots, d, e_{p+1}, \ldots, e_n) - s_p s(d-1, \ldots, d-1, e_{p+1}, \ldots, e_n) \tag{12}$$

を考える．s_p の項は $x_{\sigma(1)} \cdots x_{\sigma(p)}$ の形であるから，この項をかけて型が $(d, \ldots, d, e_{p+1}, \ldots, e_n)$ となるのは $s(d-1, \ldots, d-1, e_{p+1}, \ldots, e_n)$ の項で $x_{\sigma(1)}, \ldots, x_{\sigma(p)}$ の指数がすべて $d-1$ となる場合である．

また，$s(d, \ldots, d, e_{p+1}, \ldots, e_n)$ の各項に対してそのような s_p と $s(d-1, \ldots, d-1, e_{p+1}, \ldots, e_n)$ の項はそれぞれただ一つであるから，式 (12) では差し引かれて $(d, \ldots, d, e_{p+1}, \ldots, e_n)$ の型の単項式は存在しない．したがって，式 (12) の項の指数が d となる変数の数は p より少ない．特に $p = 1$ であればすべての変数について指数は $d-1$ 以下となる．$p = 1$ の場合は d についての帰納法の仮定から，$p > 1$ の場合は d と p についての帰納法の仮定から，式 (12) は $s_1, \ldots, s_n$ の整数係数多項式となる．したがって $s(d, \ldots, d, e_{p+1}, \ldots, e_n)$ もそうなる．

以上で数学的帰納法により，すべての p で，したがって $e_1 = d$ でも定理が成り立つことになり証明が終わる．　証明終わり

問 7.6 3 変数の対称多項式 $s(5, 0, 0) = x^5 + y^5 + z^5$ を基本対称式 $s_1 = (x + y + z)$, $s_2 = (xy + yz + zx)$, $s_3 = (xyz)$ の多項式で表せ．

すこし難しいが次の問題を挙げておく．

問 7.7 3 変数の対称多項式 $s(e_1, e_2, e_3)$ を基本対称式 s_1, s_2, s_3 の多項式で表すプログラムを C で作成せよ．

```
//
// 計算機数学 B プログラム 4
// sym3.c
// 次数 29 までの $3$ 変数対称式を x+y+z, xy+xz+yz, xyz の多項式に表す
// TeX ファイルとして出力する
```

```
#include <stdio.h>

#define N 30
#define N2 15
#define N3 10

FILE *fp;

int main()
{
    int a[N][N2][N2+1][N3+1];
    int d, p, q, i, j, k, t, E, F, G;
    fp = fopen("sym3.tex","w");
    fputs("\\documentclass[12pt]{jarticle}\n\n\\begin{document}\n\n", fp);
    for (d=0; d<N; d++) {
        for (q=0; q<N2; q++) {
            for (j=-1; j<N2; j++) {
                for (k=-1; k<N3; k++)  a[d][q][j+1][k+1] = 0;
            }
        }
    }
    a[0][0][0+1][0+1] = 1; a[1][0][0+1][0+1] = 1;
    a[2][0][0+1][0+1] = 1; a[2][0][1+1][0+1] = -2; a[2][1][1+1][0+1] = 1;
    for (d=3; d<N; d++) {
        for (p=(d+1)/2; p<=d; p++) {
            q = d-p;
            fprintf(fp, "$s(%d,%d,0) = ", p, q);
            if (q == 0) E = 0;
            else if (q == 1 && d == 3) E = 3;
            else if (q ==1) E = 2;
            else if (q == 2 && d == 4) E = 2;
            else E = 1;
            if (2*q == d-2) F = 2;
            else if (2*q < d-2) F = 1;
            else F = 0;
            if (2*q == d) G =1; else G = 0;
            for (j=0; 2*j<=d; j++) {
                for (k=0; 2*j+3*k<=d; k++)  {
```

```
                    i = d-2*j-3*k;
                    t = a[d-1-G][q-G][j-G+1][k+1] - F*a[d][q+1][j+1][k+1]
                        - E*a[d-3][q-1-G][j+1][k-1+1];
                    if (t != 0) {
                        if (t > 1) fprintf(fp, "+%d", t);
                        else if (t == 1) fprintf(fp, "+");
                        else if (t < -1) fprintf(fp, "-%d", -t);
                        else if (t == -1) fprintf(fp, "-");
                        if (i > 1) fprintf(fp, "(x+y+z)^{%d}", i);
                        else if (i == 1) fprintf(fp, "(x+y+z)");
                        if (j > 1) fprintf(fp, "(xy+yz+zx)^{%d}", j);
                        else if (j == 1)fprintf(fp, "(xy+yz+zx)");
                        if (k > 1) fprintf(fp, "(xyz)^%d", k);
                        else if (k == 1) fprintf(fp, "(xyz)");
                    }
                    a[d][q][j+1][k+1] = t;
                }
            }
            fputs("$\n\n\\vspace{2mm}\n", fp);
        }
    }
    fputs("\\end{document}", fp);
    fclose(fp);
    return 0;
}
```

このプログラムを実行すると TeX ファイルが出力され，タイプセットを行うと $p \geq q \geq 0$, $3 \leq d = p + q \leq 29$ なるすべての (p, q) について $s(p, q, 0)$ を $x + y + z$, $xy + yz + zx$, xyz の多項式で表示する PDF が得られる．プログラム中の `a[p][q][j][k]` は

$$s(p, q, 0) = \sum_{j,k} a(p, q, j, k)(x + y + z)^{d-2j-3k}(xy + yz + zx)^j(xyz)^k$$

の係数 $a(p, q, j, k)$ である．$p - 3 \geq q \geq 2$ の場合は漸化式

$$s(p, q, 0) = s(p - 1, q, 0)(x + y + z) - s(p - 1, q + 1, 0) - s(p - 2, q - 1, 0)(xyz)$$

がありこれが基本であるが，$p \leq q + 2$ や $q \leq 1$ の場合は細かく場合分けが必要で，読みやすく書けば長くなるが，このプログラムは短くするためにパラメーター E, F, G を導入して，場合に合わせて E, F, G を変化させている．

8　ブッフベルガーの判定法

多項式環 $K[x_1,\ldots,x_n]$ のイデアル I がその生成系 $G=\{g_1,\ldots,g_s\}$ により与えられた場合，G が I のグレブナー基底であるか否かの判定は定義に従えば I の先頭項イデアル $L(I)$ を知っていることが必要となる．しかし，$L(I)$ の定義は無限集合であるイデアル I に依るので，実際に計算する場合の $L(I)$ の有限な生成系の確定はグレブナー基底の構成と同時に行われる．すなわち，$L(I)$ が不確定の段階でも $\{\mathrm{LM}(g_1),\ldots,\mathrm{LM}(g_s)\}$ で生成されるイデアル $L(G)$ が $L(I)$ に等しくないことが判定出来れば，$\mathrm{LM}(g)$ が $L(G)$ に含まれない $g\in I$ を見つけて G に追加する．追加して $L(G)$ が $L(I)$ に等しいと判定出来れば G はグレブナー基底であり $\{\mathrm{LM}(g_1),\ldots,\mathrm{LM}(g_s)\}$ が $L(I)$ の生成系である．この判定と追加を並行して効率よく行う手法であるブッフベルガーの判定法を紹介する．

2 つの単項式 x^α, x^β の最小公倍単項式 $x^\gamma=\mathrm{lcm}\{x^\alpha,x^\beta\}$ は，$\alpha=(a_1,\ldots,a_n)$, $\beta=(b_1,\ldots,b_n)$ とすれば，$c_i=\max\{a_i,b_i\}$ $(i=1,\ldots,n)$ による $\gamma=(c_1,\ldots,c_n)$ で定義される．

$f,g\in K[x_1,\ldots,x_n]\setminus\{0\}$ に対して $\mathrm{LM}(f)=x^\alpha$, $\mathrm{LM}(g)=x^\beta$, $\mathrm{lcm}\{x^\alpha,x^\beta\}=x^\gamma$ とする．このとき

$$\mathrm{LM}(x^{\gamma-\alpha}f)=\mathrm{LM}(x^{\gamma-\beta}g)=x^\gamma$$

となることがわかる．ここで

$$\mathrm{spoly}(f,g)=x^{\gamma-\alpha}f-\frac{\mathrm{LC}(f)}{\mathrm{LC}(g)}x^{\gamma-\beta}g$$

と置く．$\mathrm{spoly}(f,g)$ では x^γ の項が消えて $\mathrm{LM}(\mathrm{spoly}(f,g))<x^\gamma$ となることがわかる．

補題 8.1 $G=\{g_1,\ldots,g_s\}$ で，ある $i\neq j$ について $\overline{\mathrm{spoly}(g_i,g_j)}^G=0$ とする．単項式 x^δ が $x^\delta\geq_{\mathrm{nat}}\mathrm{LM}(g_i),\mathrm{LM}(g_j)$ を満たせば，次の条件を満たす $u_1,\ldots,u_s\in K[x_1,\ldots,x_n]$ が存在する．
(1) $u_i,u_j\neq 0$ かつ $\mathrm{LM}(u_ig_i)=\mathrm{LM}(u_jg_j)=x^\delta$.
(2) $k\neq i,j$ であれば $u_k=0$ または $\mathrm{LM}(u_kg_k)<x^\delta$.
(3) $u_1g_1+\cdots+u_sg_s=0$.

証明　$\mathrm{LM}(g_i)=x^\alpha$, $\mathrm{LM}(g_j)=x^\beta$, $\mathrm{lcm}\{x^\alpha,x^\beta\}=x^\gamma$ とすれば，$\mathrm{LM}(\mathrm{spoly}(g_i,g_j))<x^\gamma$ であり，$\mathrm{spoly}(g_i,g_j)$ に補題 6.3 とその構成法を適用すれば $f_i=0$ または $\mathrm{LM}(f_ig_i)<x^\gamma$ を満たす $f_1,\ldots,f_s$ が存在して

$$\mathrm{spoly}(g_i,g_j)=f_1g_1+\cdots+f_sg_s+\overline{\mathrm{spoly}(g_i,g_j)}^G$$

となる．仮定から $\overline{\mathrm{spoly}(g_i,g_j)}^G=0$ であるから，等式

$$x^{\gamma-\alpha}g_i-\frac{\mathrm{LC}(g_i)}{\mathrm{LC}(g_i)}x^{\gamma-\beta}g_j=f_1g_1+\cdots+f_sg_s$$

を得る．$x^\delta\geq_{\mathrm{nat}}x^\gamma$ であるから，両辺に $x^{\delta-\gamma}$ をかけて右辺にまとめることにより，

$$u_i=x^{\delta-\gamma}f_i-x^{\delta-\alpha}\ ,\ \ u_j=x^{\delta-\gamma}f_j+(\mathrm{LC}(g_i)/\mathrm{LC}(g_j)))x^{\delta-\beta}\ ,\ \ u_i=x^{\delta-\gamma}f_k\ (k\neq i,j)$$

と置けば条件を満たすことがわかる．　証明終わり

定理 8.2 (ブッフベルガー) $G = \{g_1, \ldots, g_s\}$ がイデアル I を生成しているとする．G が I のグレブナー基底となるための必要十分条件は，任意の $1 \leq i < j \leq s$ について $\overline{\mathrm{spoly}(g_i, g_j)}^G = 0$ となることである．

証明　$\mathrm{spoly}(g_i, g_j) \in I$ であるから G がグレブナー基底であれば補題 6.6 により $\overline{\mathrm{spoly}(g_i, g_j)}^G$ は 0 である．

任意の i, j について $\overline{\mathrm{spoly}(g_i, g_j)}^G = 0$ とする．G がグレブナー基底でないとすると，$f_1, \ldots, f_s \in K[x_1, \ldots, x_n]$ が存在して，$g = f_1 g_1 + \cdots + f_s g_s$ について $\mathrm{LM}(g) \notin L(G)$ となる．g は固定して，この等式を満たす $f_1, \ldots, f_s$ のうちで $x^\delta = \max\{\mathrm{LM}(f_1 g_1), \ldots, \mathrm{LM}(f_s g_s)\}$ が最小で，さらにこの x^δ について $x^\delta = \mathrm{LM}(f_i g_i)$ となる最小の i が最も大きくなるものを取る．任意の $j \neq i$ について $\mathrm{LM}(f_j g_j) < \mathrm{LM}(f_i g_i) = x^\delta$ であれば $\mathrm{LM}(g) = x^\delta \in L(G)$ となり矛盾するので，ある $j > i$ について $x^\delta = \mathrm{LM}(f_i g_i) = \mathrm{LM}(f_j g_j)$ となる．このとき $x^\delta \geq_{\mathrm{nat}} \mathrm{LM}(g_i), \mathrm{LM}(g_j)$ であるから，この i, j および x^δ に補題 8.1 を適用して得られる $u_1, \ldots, u_s$ を考える．$c = \mathrm{LC}(f_i)/\mathrm{LC}(u_i)$ として $f'_k = f_k - c u_k$ $(k = 1, \ldots, s)$ と置けば，

$$f'_1 g_1 + \cdots + f'_s g_s = f_1 g_1 + \cdots + f_s g_s + c(u_1 g_1 + \cdots + u_s g_s) = g$$

かつ $\mathrm{LM}(f'_1 g_1), \ldots, \mathrm{LM}(f'_i g_i) < x^\delta$, $\mathrm{LM}(f'_{i+1} g_{i+1}), \ldots, \mathrm{LM}(f'_s g_s) \leq x^\delta$ であるから，x^δ を最小に取ったことか，i が最大になるように $f_1, \ldots, f_s$ を取ったことに矛盾する．したがって G はグレブナー基底である．　証明終わり

定理 8.2 により $g = \overline{\mathrm{spoly}(g_i, g_j)}^G \neq 0$ なる g_i, g_j があれば G はグレブナー基底でない．$\mathrm{spoly}(g_i, g_j) \in I$ より $g \in I$ である一方 $\mathrm{LM}(g) \notin L(G)$ であるから，g を G に g_{s+1} として付け加えれば $L(G)$ は拡大する．多項式環のネーター性からイデアル $L(G)$ の拡大は無限には続かないので，この操作を有限回行うことで G がグレブナー基底となる．

参考文献

[1] G.-M. Greuel and G. Pfister, A Singular — Introduction to Commutative Algebra, Springer-Verlag Berlin Heidelberg, 2008.

[2] D. Cox, J. Little and D. O'Shea, Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, Third Edition, Springer New York, 2007.

[3] 丸山 正樹，グレブナー基底とその応用 (共立叢書・現代数学の潮流)，共立出版，2002.