

計算の実行に必要なプログラムは

http://www.math.kobe-u.ac.jp/~nakayama/gb_school_poly

にて入手できる.

—— コマンド早見表 ——

- less ファイル名 — ファイルを閲覧. q キーで終了.
- emacs ファイル名 & — ファイルを作成, 編集.
- openxm fep asir — Asir を編集機能ありで起動.

1 多面体の計算

例題 1 次の線形不等式系で定義される凸多面体 P について, `polymake` を使って, その頂点を計算せよ. (この凸多面体は Birkhoff 多面体 $P(2)$ と呼ばれるもので 2 元分割表とも関係がある. [6, 例 0.11] を参照せよ.)

$$P = \{[x_{11}, x_{12}, x_{21}, x_{22}] \in \mathbb{Q}^4 \mid x_{11}, x_{12}, x_{21}, x_{22} \geq 0, \\ x_{11} + x_{12} = 1, x_{21} + x_{22} = 1, x_{11} + x_{21} = 1, x_{12} + x_{22} = 1\}$$

解答 凸多面体 P の定義する線形不等式系を入力とする次のようなファイルを用意する. 1 行目に線形不等式系を入力とすることを示す `INEQUALITIES` を書き, 以下の行に線形不等式系を書いていく. 第 1 列目が定数項で, それ以降の列が各変数の係数を表している. 変数は $x_{11}, x_{12}, x_{21}, x_{22}$ の順に並んでいるとする. 例えば 6 行目 `1 -1 -1 0 0` は線形不等式 $1 - x_{11} - x_{12} \geq 0$ を表す. また等式を入力するには, 第 8 行目 `1 0 0 -1 -1` と 第 9 行目 `-1 0 0 1 1` の表す線形不等式 $1 - x_{21} - x_{22} \geq 0, 1 - x_{21} - x_{22} \leq 0$ のようにして, 不等号の向きだけ違う不等式を 2 つ与えて, 等式 $1 - x_{21} - x_{22} = 0$ を入力としている.

—— `polymake` 入力ファイル `birkhoff2.p` ——

```
INEQUALITIES
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
1 -1 -1 0 0
-1 1 1 0 0
1 0 0 -1 -1
-1 0 0 1 1
1 -1 0 -1 0
-1 1 0 1 0
1 0 -1 0 -1
-1 0 1 0 1
```

凸多面体 P の頂点を計算するため, 次のようなコマンドを入力する.

—— `polymake` による頂点の計算 ——

```
$ polymake birkhoff2.p VERTICES
VERTICES
1 1 0 0 1
1 0 1 1 0
```

この出力は凸多面体 P の頂点を表しており, 例えば 2 行目の `1 1 0 0 1` は, 頂点 $[1, 0, 0, 1]$ を表している. (第 1 列目は無視する.) だから凸多面体 P の頂点は $[1, 0, 0, 1], [0, 1, 1, 0]$ であり, $P = \text{CONV}(\{[1, 0, 0, 1], [0, 1, 1, 0]\})$ となる. 2×2 表の形で頂点を書けば, $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ であり, 各行, 各列に 1 が 1 回だけ現れる形になっていることに注意する.

問題 2 Birkhoff 多面体 $P(d)$ を考える.

$$P(d) = \{[x_{ij}]_{1 \leq i, j \leq d} \in \mathbb{Q}^{d^2} \mid \text{任意の } i, j \text{ について } x_{ij} \geq 0, \sum_{k=1}^d x_{ik} = 1, \sum_{k=1}^d x_{kj} = 1\}$$

Birkhoff 多面体 $P(3), P(4)$ の頂点を計算せよ. 一般の $P(d)$ の頂点はどうか. (詳しくは [4, 定理 8.6] を参照)

例題 3 次の点たちの凸包で定義される凸多面体 P について, `polymake` を使って, それを定義する線形不等式系を求めよ. (この凸多面体は置換多面体 (permutahedron) Π_2 と呼ばれるものである. [6, 例 0.10] を参照せよ.)

$$P = \text{CONV}(\{[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]\})$$

解答 これは先の例題と逆の操作である. 凸多面体 P の点を入力とする次のようなファイルを用意する. 各行の最初の成分は 1 とおき, その後に点の座標を書く. だから, 第 1 行目の 1 1 2 3 は点 $[1, 2, 3]$ に対応している.

polymake 入力ファイル permutation2.p

```
POINTS
1 1 2 3
1 1 3 2
1 2 1 3
1 2 3 1
1 3 1 2
1 3 2 1
```

凸多面体 P を定義する線形不等式系を計算するため, 次のようなコマンドを入力する.

polymake によるファセットの計算

```
$ polymake permutation2.p FACETS
FACETS
-1 0 1 0
-3 1 1 0
-1 1 0 0
3 0 -1 0
5 -1 -1 0
3 -1 0 0
```

この出力は凸多面体のファセットを定義する線形不等式系を表しており, 例えば 3 行目の $-3 \ 1 \ 1 \ 0$ は, 線形不等式 $-3 + x_1 + x_2 \geq 0$ を表している. さらに, ファイル `permutation2.p` に書かれた計算結果を見ると,

ファイル permutation2.p 中の計算結果の一部

```
AFFINE_HULL
-6 1 1 1
```

これは, 凸多面体を定義する線形不等式系の中に等式 $-6 + x_1 + x_2 + x_3 = 0$ があることを示している. だから凸多面体 P は,

$$P = \{[x_1, x_2, x_3] \mid -1 + x_2 \geq 0, -3 + x_1 + x_2 \geq 0, -1 + x_1 \geq 0, \\ 3 - x_2 \geq 0, 5 - x_1 - x_2 \geq 0, 3 - x_1 \geq 0, x_1 + x_2 + x_3 = 6\}$$

と表されることがわかり, これを図示すれば図 1 のようになる.

問題 4 置換多面体

$$\Pi_{d-1} = \text{CONV}(\{[\sigma(1), \dots, \sigma(d)] \mid \sigma \in S_d\})$$

を考える. ここで S_d は d 次対称群である. 置換多面体 Π_3, Π_4 について, 上と同じ計算を行え.

2 トーリックイデアル, イニシャルフォームイデアル, グレブナー錐, グレブナー扇

例題 5 次の配置行列 A について, トーリックイデアル I_A を計算せよ.

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

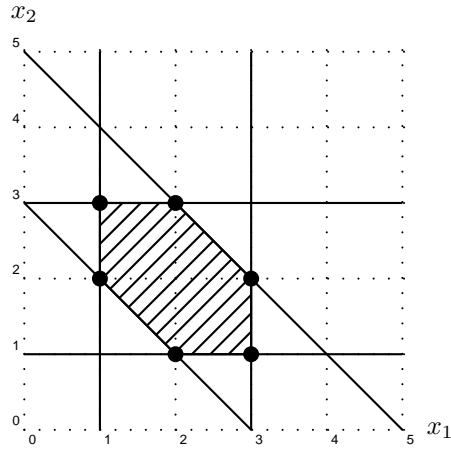


図 1: 超平面 $x_1 + x_2 + x_3 = 6$ での置換多面体 Π_2

解答 (Asir で計算)

Asir で計算させるとすれば, 次のようにすればよい. (演習用プログラム toric.rr をダウンロードして使う)

Asir での実行例

```
[1630] load("toric.rr");
[1648] A=[[0,0,1,1,1],[0,1,-1,0,1],[1,1,1,1,1]]; <- 配置行列 A の設定
[[0,0,1,1,1],[0,1,-1,0,1],[1,1,1,1,1]]
[1649] toric_ideal(A); <- トーリックイデアル I_A の計算
ideal :
[x5-t0*t1*t2,x4-t0*t2,t1*x3-t0*t2,x2-t1*t2,x1-t2,t0*t1*t2*t3-1]
gb :
[x3*x5-x4^2,x1*x5-x2*x4,x1*x4-x2*x3,-t3*x5+1,x1-t2,x5-t1*x4,x4-t1*x3,x2-t1*x1,-x5+t0*x2,x4-t0*x1]
[x3*x5-x4^2,x1*x5-x2*x4,x1*x4-x2*x3]
```

トーリックイデアルは $I_A = \langle x_3x_5 - x_4^2, x_1x_5 - x_2x_4, x_1x_4 - x_2x_3 \rangle$ であることがわかる.

(4ti2 で計算)

4ti2 で計算させるとすれば, まず配置行列を書いた次のような入力ファイル example1 を用意する.

入力ファイル example1

```
3 5
0 0 1 1 1
0 1 -1 0 1
1 1 1 1 1
```

4ti2 のコマンド groebner を実行すれば I_A のグレブナー基底を計算して, その計算結果を example1.gro に出力してくれる.

4ti2 コマンド groebner の実行

```
$ groebner example1
```

出力ファイル example1.gro

```
3 5
-1 1 0 1 -1
-1 1 1 -1 0
0 0 -1 2 -1
```

問題 6 次の配置行列 A について, トーリックイデアル I_A を計算せよ. (計算機で)

$$A_3 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, A_4 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

実は素手で計算できる (詳細は [2] を参照).

例題 7 イデアル

$$I = \langle x^2 + y - 1, x + y^2 - 1 \rangle$$

について, 各重みベクトル

$$\mathbf{w}_1 = [3, 1], \mathbf{w}_2 = [1, 1], \mathbf{w}_3 = [1, 3], \mathbf{w}_4 = [2, 1], \mathbf{w}_5 = [1, 2], \mathbf{w}_6 = [0, 0]$$

についてのイニシャルフォームイデアル $\text{in}_{\mathbf{w}}(I)$ を計算せよ.

解答 単項式順序 $<_{\mathbf{w}}$ についてのグレブナー基底 \mathcal{G} とすると, $\text{in}_{\mathbf{w}}(\mathcal{G}) = \{\text{in}_{\mathbf{w}}(g) \mid g \in \mathcal{G}\}$ が $\text{in}_{\mathbf{w}}(I)$ の単項式順序 $<$ についてのグレブナー基底となる. 単項式順序 $<_{\mathbf{w}_i}$ (ここでタイブレイクの $<$ は逆辞書式順序とする) についてのグレブナー基底 \mathcal{G}_i を計算すれば

$$\mathcal{G}_1 = \{x + y^2 - 1, -y^4 + 2y^2 - y\}$$

$$\mathcal{G}_2 = \{x + y^2 - 1, x^2 + y - 1\}$$

$$\mathcal{G}_3 = \{x^2 + y - 1, -x^4 + 2x^2 - x\}$$

$$\mathcal{G}_4 = \{x^2 + y - 1, x + y^2 - 1\}$$

$$\mathcal{G}_5 = \{x^2 + y - 1, x + y^2 - 1\}$$

$$\mathcal{G}_6 = \{x^2 + y - 1, x + y^2 - 1\}$$

となるので, 各グレブナー基底について, 重みベクトルに関するイニシャルフォームをとってやれば,

$$\text{in}_{\mathbf{w}_1}(I) = \langle x, -y^4 \rangle$$

$$\text{in}_{\mathbf{w}_2}(I) = \langle y^2, x^2 \rangle$$

$$\text{in}_{\mathbf{w}_3}(I) = \langle y, -x^4 \rangle$$

$$\text{in}_{\mathbf{w}_4}(I) = \langle x^2, x + y^2 \rangle$$

$$\text{in}_{\mathbf{w}_5}(I) = \langle x^2 + y, y^2 \rangle$$

$$\text{in}_{\mathbf{w}_6}(I) = \langle x^2 + y - 1, x + y^2 - 1 \rangle$$

となる.

単項式順序 $<_{\mathbf{w}}$ についてのグレブナー基底を Asir で計算させるには, 例えば付属プログラム `toric.rr` 中の関数 `gr_w` を用いて次のようにすればよい. 関数 `gr_w(Id, VL, W)` は, 変数リストが `VL` のイデアル `Id` の重みベクトル `W` に付随する単項式順序 $<_{\mathbf{w}}$ (ここでタイブレイクの $<$ は逆辞書式順序) についてのグレブナー基底を返す. また関数 `in_w(P, VL, W)` により, 多項式 `P` の重みベクトル `W` についてのイニシャルフォーム $\text{in}_{\mathbf{w}}(P)$ を計算している.

```

[1358] load("toric.rr");           プログラム toric.rr を読み込み
[1376] Id=[x+y^2-1, x^2+y-1];
[1377] VL=[x,y];
[1378] G1=gr_w(Id,VL,[3,1]);       Id の <_[3,1] についてのグレブナー基底
[x+y^2-1,-y^4+2*y^2-y]
[1379] IN1 = map(in_w, G1, VL, [3,1]); G1 の各元のイニシャルフォームをとる
[x,-y^4]
[1380] G2=gr_w(Id,VL,[1,1]);       Id の <_[1,1] についてのグレブナー基底
[x+y^2-1,x^2+y-1]
[1381] IN2 = map(in_w, G2, VL, [1,1]); G2 の各元のイニシャルフォームをとる
[y^2,x^2]
[1382] G3=gr_w(Id,VL,[1,3]);       Id の <_[1,3] についてのグレブナー基底
[x^2+y-1,-x^4+2*x^2-x]
[1383] IN3 = map(in_w, G3, VL, [1,3]); G3 の各元のイニシャルフォームをとる
[y,-x^4]
[1384] G4=gr_w(Id,VL,[2,1]);       Id の <_[2,1] についてのグレブナー基底
[x+y^2-1,x^2+y-1]
[1385] IN4 = map(in_w, G4, VL, [2,1]); G4 の各元のイニシャルフォームをとる
[x+y^2,x^2]
[1387] G5=gr_w(Id,VL,[1,2]);       Id の <_[1,2] についてのグレブナー基底
[x^2+y-1,x+y^2-1]
[1388] IN5 = map(in_w, G5, VL, [1,2]); G5 の各元のイニシャルフォームをとる
[x^2+y,y^2]
[1390] G6=gr_w(Id,VL,[0,0]);       Id の <_[0,0] についてのグレブナー基底
[x+y^2-1,x^2+y-1]
[1391] IN6 = map(in_w, G6, VL, [0,0]); G6 の各元のイニシャルフォームをとる
[x+y^2-1,x^2+y-1]

```

問題 8 イデアル

$$I = \langle x + y + z, xy + yz + zx, xyz - 1 \rangle$$

について、各重みベクトル

$$\mathbf{w}_1 = [3, 2, 1], \mathbf{w}_2 = [3, 1, 2], \mathbf{w}_3 = [2, 1, 3], \mathbf{w}_4 = [1, 2, 3], \mathbf{w}_5 = [2, 3, 1], \mathbf{w}_6 = [1, 3, 2]$$

についてのイニシャルフォームイデアル $\text{in}_w(I)$ を計算せよ。

例題 9 (Gfan を用いた普遍グレブナー基底の計算)

例題 7 で出てきたイデアル

$$I = \langle x^2 + y - 1, x + y^2 - 1 \rangle$$

について、狭義普遍グレブナー基底を Gfan を用いて計算せよ。

解答 狭義普遍グレブナー基底を計算させるソフトウェアとして Gfan がある。これを用いて狭義普遍グレブナー基底を計算させる。イデアルの生成元を書いた次の入力ファイルを用意する。

Gfan 入力ファイル gfan_input.txt

```

Q[x,y]
{x^2+y-1, x+y^2-1}

```

最初の $Q[x,y]$ は計算対象の環を表し、生成元たちは中括弧 $\{ \}$ でくる。このファイルを入力として、コマンド gfan を次のように実行する。

Gfan のコマンド gfan の実行結果

```

$ gfan < gfan_input.txt
Q[x,y]
LP algorithm being used: "cddgmp".
{
{
y^4+y-2*y^2,
x-1+y^2}
,
{
y^2-1+x,
x^2-1+y}
,
{
y-1+x^2,
x^4+x-2*x^2}
}

```

ここで出力されている 3 つの集合がそれぞれ被約グレブナー基底であり、各多項式で先頭に表示されているものが先頭項

($\text{in}_{<}$) である. たとえば, 一番目の出力 $\{y^4 + y - 2xy^2, x - 1 + y^2\}$ は, 被約グレブナー基底で先頭項が y^4, x であるものを表している. この 3 つの集合の和集合が狭義普遍グレブナー基底となっている. また例題 7 で計算した G_1, G_2, G_3 が, この 3 つの集合に一致していることが確認できる.

問題 10 $I = \langle x + y + z, xy + yz + zx, xyz - 1 \rangle$ について, 狭義普遍グレブナー基底を *Gfan* を用いて計算せよ.

例題 11 イデアル $I = \langle x^2 + y^2 - 4, xy - 1 \rangle$ について, 次の計算を行う.

1. $x > y$ なる純辞書式順序 $<_{\text{purelex}}$ に対して, グレブナー錐 $C[<_{\text{purelex}}] = \{\mathbf{w}' \in \mathbb{Q}_{\geq 0}^2 \mid \text{in}_{\mathbf{w}'}(I) = \text{in}_{<_{\text{purelex}}}(I)\}$ を求めよ.
2. 重みベクトル $\mathbf{w} = [3, 1]$ について, グレブナー錐 $C[\mathbf{w}] = \{\mathbf{w}' \in \mathbb{Q}_{\geq 0}^2 \mid \text{in}_{\mathbf{w}'}(I) = \text{in}_{[3,1]}(I)\}$ を計算せよ.
3. イデアル I の $\mathbb{Q}_{\geq 0}^2$ におけるグレブナー扇 $\text{GF}(I)$ を計算し, 図示せよ.
4. グレブナー扇 $\text{GF}(I)$ の各錐における (被約) グレブナー基底を計算し, $\text{in}_{\mathbf{w}}(I)$ を求めよ.

解答 1. イデアル I の $<_{\text{purelex}}$ に関する被約グレブナー基底は

$$\{g_1 = \underline{y^4} - 4y^2 + 1, g_2 = x + y^3 - 4y\}$$

である. (下線部は先頭項を表す.)

Asir での計算結果

```
[1659] Id = [x^2+y^2-4, x*y-1];
[x^2+y^2-4, y*x-1]
[1660] GB = nd_gr(Id, [x,y], 0, 2);
[y^4-4*y^2+1, -x-y^3+4*y]
```

条件

$$\text{in}_{\mathbf{w}}(g_1) = \text{in}_{<_{\text{purelex}}}(g_1) = y^4, \quad \text{in}_{\mathbf{w}}(g_2) = \text{in}_{<_{\text{purelex}}}(g_2) = x$$

より, 重みベクトル $\mathbf{w} = [w_1, w_2]$ に関する次の線形不等式系を得る.

$$\begin{aligned} [0, 2] \cdot \mathbf{w} > 0, & \quad [0, 4] \cdot \mathbf{w} > 0, \\ [1, -3] \cdot \mathbf{w} > 0, & \quad [1, -1] \cdot \mathbf{w} > 0, \\ [1, 0] \cdot \mathbf{w} > 0, & \quad [0, 1] \cdot \mathbf{w} > 0 \end{aligned}$$

例えば線形不等式 $[1, -3] \cdot \mathbf{w} > 0$ は, 条件 $\text{in}_{\mathbf{w}}(g_2) = x$ より得られる. 具体的に言うと, 多項式 g_2 において, x の \mathbf{w} についての重み $[1, 0] \cdot \mathbf{w}$ が, y^3 の \mathbf{w} についての重み $[0, 3] \cdot \mathbf{w}$ よりも大きいことから導かれたものである. この線形不等式系を解けば, $C[<_{\text{purelex}}] = \{\mathbf{w} \mid w_2 > 0, w_2 < 1/3w_1\}$ が得られる.

上の線形不等式系を *polymake* で計算するには, 線形不等式系を表す以下のファイル

polymake 入力ファイル cone1.p

```
INEQUALITIES
0 0 2
0 0 4
0 1 -3
0 1 -1
0 1 0
0 0 1
```

を用意して, 次のコマンドを実行する.

polymake によるファセットの計算

```
$ polymake cone1.p FACETS
FACETS
0 0 2
0 1 -3
1 0 0
```

この出力より, $2w_2 > 0, w_1 - 3w_2 > 0, 1 > 0$ が得られる.

2. イdeal I の $\prec_{[3,1]}$ に関する被約グレブナー基底は

$$\{g_1 = \underline{y^4} - 4y^2 + 1, g_2 = \underline{x + y^3} - 4y\}$$

である。(下線部は重みベクトル $[3, 1]$ についてのイニシャルフォームを表す。) 条件

$$\text{in}_{\mathbf{w}}(g_1) = \text{in}_{[3,1]}(g_1) = y^4, \quad \text{in}_{\mathbf{w}}(g_2) = \text{in}_{[3,1]}(g_2) = x + y^3$$

より, $\mathbf{w} = [w_1, w_2]$ に関する次の線形不等式系が得られる.

$$\begin{aligned} [1, -3] \cdot \mathbf{w} &= 0, & [1, -1] \cdot \mathbf{w} &> 0, \\ [0, 2] \cdot \mathbf{w} &> 0, & [0, 4] \cdot \mathbf{w} &> 0, \\ [1, 0] \cdot \mathbf{w} &> 0, & [0, 1] \cdot \mathbf{w} &> 0 \end{aligned}$$

例えば線形不等式 $[1, -3] \cdot \mathbf{w} = 0$ は, 条件 $\text{in}_{\mathbf{w}}(g_2) = x + y^3$ から得られる. 具体的に言うと, 多項式 g_2 において, x の \mathbf{w} についての重み $[1, 0] \cdot \mathbf{w}$ が, y^3 の \mathbf{w} についての重み $[0, 3] \cdot \mathbf{w}$ と等しいことから導かれたものである. この線形不等式系を解けば, $C[[3, 1]] = \{[w_1, w_2] \mid w_2 > 0, w_2 = 1/3w_1\}$.

polymake で計算するには以下のファイル

polymake 入力ファイル cone2.p

```
INEQUALITIES
0 1 -3
0 -1 3
0 1 -1
0 0 2
0 0 4
0 1 0
0 0 1
```

を用意して, 次のコマンドを実行する. ここで, データの 1 行目 $0 \ 1 \ -3$ と 2 行目 $0 \ -1 \ 3$ (すなわち, 線形不等式 $w_1 - 3w_2 \geq 0, -w_1 + 3w_2 \geq 0$) で, 等式 $w_1 = \frac{1}{3}w_2$ を表現していることに注意する.

polymake によるファセット, affine_hull の計算

```
$ polymake cone2.p FACETS AFFINE_HULL
FACETS
0 1 -1
1 0 0

AFFINE_HULL
0 1 -3
```

この出力より, $w_1 - w_2 > 0, w_1 - 3w_2 = 0$ がわかる.

3. $\overline{C[\prec_{\text{purelex}}]}$ に属さないが十分近い \mathbf{w} を一つとる. 今, 先にやった 2. より, $[3, 1]$ が境界上の点であることが分かっているので, $\mathbf{w} = [3, 2]$ を取ってみる. このとき, $C[\mathbf{w}]$ を計算すると,

$$C[\mathbf{w}] = \{[w_1, w_2] \mid w_2 < w_1, w_2 > 1/3w_1\}$$

であり, $C[\prec_{\text{purelex}}]$ と境界を共有している. (境界を共有しなかった場合はもっと近くの \mathbf{w} を取り直す.) これを $\mathbb{Q}_{\geq 0}^2$ を尽くすまで繰り返す. 今は x と y の対称性から残りの部分はすぐに分かる. 図示すると図 2 のようになり, $\text{GF}(I)$ は 10 個の元から成ることが分かる.

4. 各錐 C_i に属する \mathbf{w}_i を用いて $\prec_{\mathbf{w}_i}$ (タイブレークの \prec は純辞書式順序) に関する被約グレブナー基底を計算する. 下線部が生成元のイニシャルフォームである.

- $C_1 : \{x + y^3 - 4y, \underline{y^4} - 4y^2 + 1\}$
- $C_2 : \{\underline{x^2} + y^2 - 4, \underline{xy} - 1, \underline{y^3} + x - 4y\}$
- $C_3 : \{\underline{y^2} + x^2 - 4, \underline{xy} - 1, \underline{x^3} - 4x + y\}$
- $C_4 : \{\underline{x^4} - 4x^2 + 1, \underline{y} + x^3 - 4x\}$

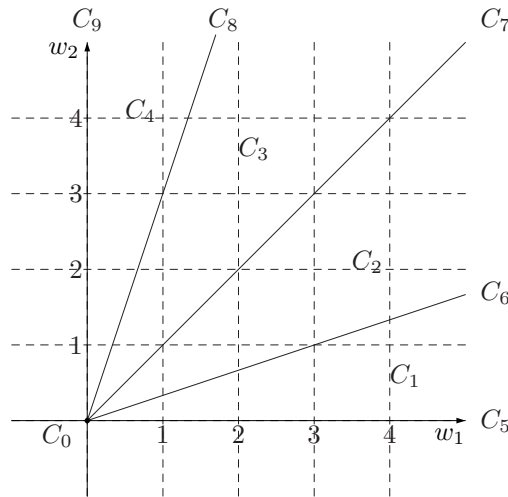


図 2: I のグレブナー扇

- $C_5 : \{\underline{x + y^3 - 4y}, \underline{y^4 - 4y^2 + 1}\}$
- $C_6 : \{\underline{x + y^3 - 4y}, \underline{y^4 - 4y^2 + 1}\}$
- $C_7 : \{\underline{x^2 + y^2 - 4}, \underline{xy - 1}, \underline{y^3 + x - 4y}\}$
- $C_8 : \{\underline{y^2 + x^2 - 4}, \underline{xy - 1}, \underline{x^3 + y - 4x}\}$
- $C_9 : \{\underline{x^4 - 4x^2 + 1}, \underline{y + x^3 - 4x}\}$
- $C_0 : \{\underline{x + y^3 - 4y}, \underline{y^4 - 4y^2 + 1}\}$

問題 12 イデアル $I = \langle x + y - 1, xy - 1 \rangle$ について,

1. $x > y$ なる純辞書式順序 \prec_{purelex} について, グレブナー錐 $C[\prec_{\text{purelex}}]$ を計算せよ.
2. $y > x$ なる純辞書式順序 \prec'_{purelex} について, グレブナー錐 $C[\prec'_{\text{purelex}}]$ を計算せよ.
3. $\mathbb{Q}_{\geq 0}^2$ におけるグレブナー扇 $\text{GF}(I)$ を計算し, 図示せよ.

3 正則三角形分割

例題 13

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

について, 以下を手計算せよ.

1. トーリックイデアル I_A を求めよ.
2. $\mathbf{w} = [2, 1, 2]$ について $\text{in}_{\mathbf{w}}(I_A)$ を求めよ.
3. $\mathbf{w} = [2, 1, 2]$ について, 正則三角形分割 $\Delta_{\mathbf{w}}$ を求めよ.
4. $\mathbf{w} = [1, 2, 1]$ について, 正則三角形分割 $\Delta_{\mathbf{w}}$ を求めよ.

解答 1. $I_A = \langle x_0x_2 - x_1^2 \rangle$

2. $\text{in}_{\mathbf{w}}(I_A) = \langle x_0x_2 \rangle$

- $\sqrt{\text{in}_w(I_A)} = \langle x_0x_2 \rangle = \langle x_0 \rangle \cap \langle x_2 \rangle$ である。系 5.5.7 より、正則三角形分割 Δ_w の極大単体は 12 と 01 とわかる。ここで、12 と 01 は単体 $\text{CONV}(\{a_1, a_2\})$ と $\text{CONV}(\{a_0, a_1\})$ を表すものとする。
- $\text{in}_w(I_A) = \langle x_1^2 \rangle$ であり、 $\sqrt{\text{in}_w(I_A)} = \langle x_1 \rangle$ となるから、正則三角形分割 Δ_w の極大単体は 02。

問題 14

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

について、以下を手計算せよ。

- トーリックイデアル I_A を求めよ。
- $w = [1, 0, 0, 2]$ について $\text{in}_w(I_A)$ を求めよ。
- 上の w について、正則三角形分割 Δ_w を求めよ。

例題 15 配置行列 (2×3 分割表に対応する配置行列)

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

を考える。

- トーリックイデアル I_A の生成系を求めよ。
- トーリックイデアル I_A のグレブナー扇を求めよ。
- A の全ての正則三角形分割を列挙せよ。

解答 1. I_A の生成系として、

$$I_A = \langle -x_2x_6 + x_3x_5, -x_1x_6 + x_3x_4, -x_1x_5 + x_2x_4 \rangle$$

がとれる。

- Gfan を用いてグレブナー扇を計算させる。イデアル I_A の生成系を入力とするファイルを作る。

————— Gfan 入力ファイル gfan_cont2x3.txt —————

```
Q[x1,x2,x3,x4,x5,x6]
{-x2*x6+x3*x5,-x1*x6+x3*x4,-x1*x5+x2*x4}
```

gfan コマンドを実行し、出力を gfan_cont2x3.out に出す。

————— Gfan の実行 —————

```
$ gfan < gfan_cont2x3.txt > gfan_cont2x3.out
```

全ての被約グレブナー基底 (全部で 6 個ある) を出力してくれる。その被約グレブナー基底たちから、グレブナー錐 $C[w]$ を計算するには、Gfan のコマンド `gfan_groebnercone` を用いる。例えば、先ほど計算した Gfan 出力ファイル `gfan_cont2x3.out` から被約グレブナー基底の 1 つである $\{x_3x_4 - x_1x_6, x_2x_6 - x_3x_5, x_2x_4 - x_1x_5\}$ を取り出し、これのグレブナー錐 $C[w]$ を計算する。まず入力ファイル `cont2x3_6.gb` を用意する。

Gfan コマンド gfan_groebnercone 入力ファイル cont2x3_6.gb

```
Q[x1,x2,x3,x4,x5,x6]
{x3*x4-x1*x6, x2*x6-x3*x5, x2*x4-x1*x5}
```

コマンド gfan_groebnercone を次のように実行する.

Gfan コマンド gfan_groebnercone によるグレブナー錐の計算

```
$ gfan_groebnercone < cont2x3_6.gb
LP algorithm being used: "cddgmp".
_application PolyhedralCone
_version 2.2
_type PolyhedralCone
```

```
AMBIENT_DIM
6
```

```
DIM
6
```

```
IMPLIED_EQUATIONS
```

```
LINEALITY_DIM
4
```

```
LINEALITY_SPACE
1 0 0 0 -1 -1
0 1 0 0 1 0
0 0 1 0 0 1
0 0 0 1 1 1
```

```
FACETS
-1 0 1 1 0 -1
0 1 -1 0 -1 1
```

```
RELATIVE_INTERIOR_POINT
-1 1 0 0 0 0
```

RELATIVE_INTERIOR_POINT が、グレブナー錐 $C[\mathbf{w}]$ の内点であり、 $\mathbf{w} = [-1, 1, 0, 0, 0, 0]$ ととれる。FACETS の 2 つのベクトルが、グレブナー錐 $C[\mathbf{w}]$ を定義する各超平面の法線ベクトルを表している。すなわち、このグレブナー錐 $C[\mathbf{w}]$ を定義する線形不等式系は

$$\begin{aligned} -w_1 + w_3 + w_4 - w_6 &> 0, \\ w_2 - w_3 - w_5 + w_6 &> 0. \end{aligned}$$

他の被約グレブナー基底についても、同様にグレブナー錐を計算してやる。それら閉包がグレブナー扇の最大次元の錐となる。

各グレブナー錐の内点 \mathbf{w}_i とそれに対応する被約グレブナー基底 \mathcal{G}_i は次のようになる。ここで、各 2 項式の最初に表示されている項が先頭項であり、 \mathbf{w}_i のイニシャルフォームに対応する項である。

- $\mathcal{G}_5 = \{x_3x_4 - x_1x_6, x_2x_6 - x_3x_5, x_2x_4 - x_1x_5\}$, $\mathbf{w}_5 = [-1, 1, 0, 0, 0, 0]$

3. 先の 2. の計算結果より、イニシャルフォームイデアル $\text{in}_{\mathbf{w}}(I_A)$ で単項式イデアルであるものが全てわかる。

- $\text{in}_{\mathbf{w}_5}(I_A) = \langle x_3x_4, x_2x_6, x_2x_4 \rangle$, $\mathbf{w}_5 = [-1, 1, 0, 0, 0, 0]$

イニシャルフォームイデアルに対して、1. で行ったように正則三角形分割を計算する。

```
[1631] import("noropd.rr");
[1815] I=[x3*x4,x2*x6,x2*x4];
[x3*x4,x2*x6,x2*x4]
[1816] V=[x1,x2,x3,x4,x5,x6];
[x1,x2,x3,x4,x5,x6]
[1817] noropd.prime_dec(I,V);
[0] [1] [2] [3] [pint,0,rpint,0.00212073]
[[x2,x3],[x2,x4],[x4,x6]]
```

各重みベクトル w_i について、正則三角形分割 Δ_{w_i} の極大単体は以下ようになる。

- 1456, 1356, 1235, $w_5 = [-1, 1, 0, 0, 0, 0]$

問題 16 例題 15 の Gfan の他の出力に対して同様の計算を行い、すべての正則三角形分割を列挙せよ。

例題 17 配置行列

$$A = \begin{bmatrix} 4 & 0 & 0 & 2 & 1 & 1 \\ 0 & 4 & 0 & 1 & 2 & 1 \\ 0 & 0 & 4 & 1 & 1 & 2 \end{bmatrix} = [a_0, \dots, a_5]$$

に関して、全ての三角形分割を列挙せよ。

解答 (TOPCOM による三角形分割の列挙)

TOPCOM を使えば、頂点のデータから全ての三角形分割を列挙することができる。まず、以下のような配置行列 A の各列ベクトルを並べた入力ファイル `reg_tri_1.top` を用意する。

TOPCOM 入力ファイル `reg_tri_1.top`

```
[[4,0,0],
 [0,4,0],
 [0,0,4],
 [2,1,1],
 [1,2,1],
 [1,1,2]]
```

コマンド `points2nalltriangs` を用いて全ての三角形分割の個数を数えてみよう。次いで、コマンド `points2triangs` で `bistellar flip` と呼ばれる操作 (三角形分割から別の三角形分割を作る操作、詳しくは [1] を参照) で正則三角形分割と連結な三角形分割を全て列挙する。ここで、`flip` 操作で移り合う三角形分割どうしを連結と呼んでいる。ともに、入力はファイル `reg_tri_1.top` をリダイレクトで与える。

TOPCOM による三角形分割の列挙

```
$ points2nalltriangs < reg_tri_1.top
Evaluating Commandline Options ...
... done.
18

$ points2triangs < reg_tri_1.top
Evaluating Commandline Options ...
... done.
T[1]:=[6,3:{{0,1,2}}];
T[2]:=[6,3:{{1,2,4},{0,2,4},{0,1,4}}];
T[3]:=[6,3:{{1,2,5},{0,2,5},{0,1,5}}];
T[4]:=[6,3:{{1,2,3},{0,2,3},{0,1,3}}];
T[5]:=[6,3:{{1,2,4},{0,2,3},{0,1,3},{2,3,4},{1,3,4}}];
T[6]:=[6,3:{{1,2,5},{0,2,3},{0,1,3},{2,3,5},{1,3,5}}];
T[7]:=[6,3:{{1,2,4},{0,1,4},{0,2,3},{2,3,4},{0,3,4}}];
T[8]:=[6,3:{{1,2,4},{0,1,4},{0,2,5},{2,4,5},{0,4,5}}];
T[9]:=[6,3:{{1,2,5},{0,2,5},{0,1,3},{1,3,5},{0,3,5}}];
T[10]:=[6,3:{{0,1,4},{1,2,5},{0,2,5},{0,4,5},{1,4,5}}];
T[11]:=[6,3:{{1,2,4},{0,1,4},{0,2,5},{0,3,4},{2,4,5},{0,3,5},{3,4,5}}];
T[12]:=[6,3:{{1,2,5},{0,2,5},{0,1,3},{1,3,4},{0,3,5},{1,4,5},{3,4,5}}];
T[13]:=[6,3:{{1,2,4},{0,2,3},{0,1,3},{1,3,4},{2,3,5},{2,4,5},{3,4,5}}];
T[14]:=[6,3:{{1,2,5},{0,2,3},{0,1,3},{1,3,4},{2,3,5},{1,4,5},{3,4,5}}];
T[15]:=[6,3:{{1,2,4},{0,1,4},{0,2,3},{2,3,5},{0,3,4},{2,4,5},{3,4,5}}];
T[16]:=[6,3:{{0,1,4},{1,2,5},{0,2,5},{0,3,4},{0,3,5},{1,4,5},{3,4,5}}];
T[17]:=[6,3:{{0,1,4},{1,2,5},{0,2,3},{2,3,5},{0,3,4},{1,4,5},{3,4,5}}];
T[18]:=[6,3:{{1,2,4},{0,2,5},{0,1,3},{1,3,4},{2,4,5},{0,3,5},{3,4,5}}];
```

この出力から、全ての三角形分割の個数は 18 個であり、連結なものも 18 個得られているので、三角形分割はこれで尽きる。例えば、 $\{0, 1, 3\}$ は入力ファイルの 0 行目、1 行目、3 行目の表す頂点の凸閉包が三角形分割に含まれる単体であることを表している。(インデックスが 0 から始まることに注意。) 今、 $T[13]$ が上で計算した三角形分割に対応していることが分かる。

余談であるが、`points2alltriangs` と `points2triangs` の出力が一致しないような例(つまり、正則三角形分割と連結でない三角形分割をもつような配置行列)は F. Santos [3] によって与えられた。

コマンド `points2triangs` に対しては、

- `--regular`: 正則三角形分割のみを出力する
- `--nonregular`: 非正則三角形分割のみを出力する
- `--heights`: 各三角形分割を与える高さベクトルの代表元も出力する

等のオプションが用意されている。三角形分割を実現する高さベクトルが存在すれば、その三角形分割は正則であるから、`--heights` には `--regular` が付加されていると考えてよい。例えば、次のように実行してみるとよい。

TOPCOM コマンド `points2triangs` の各種オプションの指定方法

```
$ points2triangs --regular < reg_tri_1.top
$ points2triangs --heights < reg_tri_1.top
$ points2triangs --nonregular < reg_tri_1.top
```

また、対称性を持った凸多面体に対して三角形分割を計算させるときには、入力に対称性の情報(対称群の生成元)を付加してやることで、無駄な計算を省くことができる。更に、出力も簡約化された状態で得ることができる。この例の場合は、012 と 345 に 3 次対称群 S_3 が作用するので、

TOPCOM 入力ファイル `reg_tri_1_sym.top`

```
[[4,0,0],
 [0,4,0],
 [0,0,4],
 [2,1,1],
 [1,2,1],
 [1,1,2]]

[[1,2,0,4,5,3],
 [1,0,2,4,3,5]]
```

の様に点データの下に各点の移り先を頂点のラベルで指定する。 $[1, 2, 0, 4, 5, 3]$ は反時計回りに 120° 回転の対称性を表し、 $[1, 0, 2, 4, 3, 5]$ は左右の対称性を表している。この入力ファイルに対して三角形分割を計算すると次のようになる。

TOPCOM による三角形分割の列挙

```
$ points2triangs < reg_tri_1_sym.top
Evaluating Commandline Options ...
... done.
T[1]:=[6,3:{{0,1,2}}];
T[2]:=[6,3:{{1,2,5},{0,2,5},{0,1,5}}];
T[3]:=[6,3:{{1,2,5},{0,2,5},{0,1,4},{1,4,5},{0,4,5}}];
T[4]:=[6,3:{{1,2,5},{0,2,5},{0,1,4},{1,4,5},{0,3,4},{0,3,5},{3,4,5}}];
T[5]:=[6,3:{{1,2,5},{0,1,4},{0,2,3},{1,4,5},{0,3,4},{2,3,5},{3,4,5}}];
```

問題 18 例題 15 に対して TOPCOM を使ってすべての正則三角形分割を計算せよ。

4 サークット, Graver 基底

例題 19 配置行列 (3×3 分割表に対応する配置行列)

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

について, そのトーリックイデアルの生成系 I_A , サークット集合 C_A , Graver 基底 Gr_A を計算せよ. (4ti2 を用いて)

解答 配置行列 A を表す入力ファイルとして,

4ti2 入力ファイル cont3x3

```
6 9
1 1 1 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0
0 0 0 0 0 1 1 1 1
1 0 0 1 0 0 1 0 0
0 1 0 0 1 0 0 1 0
0 0 1 0 0 1 0 0 1
```

を用意する. ここで 1 行目の 6, 9 は, 入力の行列が 6×9 の行列であることを表し, それ以降に行列を書く.

4ti2 でトーリックイデアル I_A の生成系 (グレブナー基底) を計算するには, コマンド `groebner` を用いる.

4ti2 コマンド `groebner` の実行

```
$ groebner cont3x3
```

出力はファイル `cont3x3.gro` に出力される.

4ti2 コマンド `groebner` の出力ファイル `cont3x3.gro`

```
9 9
-1 0 1 0 0 0 1 0 -1
-1 0 1 1 0 -1 0 0 0
-1 1 0 0 0 0 1 -1 0
-1 1 0 1 -1 0 0 0 0
0 -1 1 0 0 0 0 1 -1
0 -1 1 0 1 -1 0 0 0
0 0 0 -1 0 1 1 0 -1
0 0 0 -1 1 0 1 -1 0
0 0 0 0 -1 1 0 1 -1
```

この出力は, 各行ベクトルが 2 項式に対応しており, たとえば 1 行目の $-1 0 1 0 0 0 1 0 -1$ は 2 項式 $x_3x_7 - x_1x_9$ に対応する. (ただし, ここで i 列に対応する変数を x_i とおいた.) これら 9 つの 2 項式が I_A の被約グレブナー基底である. 実は 2 元分割表に対応するトーリックイデアルのグレブナー基底は, 形が知られており, 上のような計算をしなくてもわかる. 詳しくは [5, Prop 5.4]などを参照してほしい.

4ti2 でトーリックイデアル I_A のサーキット集合 C_A を計算するには, コマンド `circuits` を用いる.

4ti2 コマンド `circuits` の実行

```
$ circuits cont3x3
```

出力はファイル `cont3x3.cir` に出力される.

```

15 9
0 0 0 0 1 -1 0 -1 1
0 0 0 1 -1 0 -1 1 0
0 0 0 1 0 -1 -1 0 1
0 1 -1 -1 0 1 1 -1 0
0 1 -1 0 -1 1 0 0 0
0 1 -1 0 0 0 0 -1 1
0 1 -1 1 -1 0 -1 0 1
1 -1 0 -1 0 1 0 1 -1
1 -1 0 -1 1 0 0 0 0
1 -1 0 0 0 0 -1 1 0
1 -1 0 0 1 -1 -1 0 1
1 0 -1 -1 0 1 0 0 0
1 0 -1 -1 1 0 0 -1 1
1 0 -1 0 -1 1 -1 1 0
1 0 -1 0 0 0 -1 0 1

```

出力の読み方は、先のグレブナー基底の場合と同様である。

4ti2 でトーリックイデアル I_A の Graver 基底を計算するには、コマンド `graver` を用いる。

4ti2 コマンド `graver` の実行

```
$ graver cont3x3
```

出力はファイル `cont3x3.gra` に出力される。この場合、サーキット集合と Graver 基底は同じである。

問題 20 配置行列

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

について、そのトーリックイデアルの生成系 I_A 、サーキット集合 \mathcal{C}_A 、Graver 基底 Gr_A を計算せよ。(4ti2 を用いて。)

例題 21 配置行列

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 10 \end{bmatrix}$$

について、 A の Lawrence 持ち上げ $\Lambda(A)$ に関するトーリックイデアル $I_{\Lambda(A)}$ のグレブナー基底を求めることにより、 A の Graver 基底を計算せよ。(すなわち、定理 5.4.11 を用いた Graver 基底の計算法)

解答 4ti2 の `groebner` コマンドを使うことにする。(勿論、Asir など直接グレブナー基底を計算してもよい。) Lawrence 持ち上げた配置行列 $\Lambda(A)$ を表すファイルを用意する。

4ti2 入力ファイル `lawrence1`

```

6 8
1 1 1 1 0 0 0 0
0 1 2 10 0 0 0 0
1 0 0 0 1 0 0 0
0 1 0 0 0 1 0 0
0 0 1 0 0 0 1 0
0 0 0 1 0 0 0 1

```

`groebner` コマンドで、 $I_{\Lambda(A)}$ のグレブナー基底を計算させると、

4ti2 コマンド `groebner` の出力ファイル `lawrence1.gro`

```

11 8
-9 10 0 -1 9 -10 0 1
-8 8 1 -1 8 -8 -1 1
-7 6 2 -1 7 -6 -2 1
-6 4 3 -1 6 -4 -3 1
-5 2 4 -1 5 -2 -4 1
-4 0 5 -1 4 0 -5 1
-3 -2 6 -1 3 2 -6 1
-2 -4 7 -1 2 4 -7 1
-1 -6 8 -1 1 6 -8 1
-1 2 -1 0 1 -2 1 0
0 -8 9 -1 0 8 -9 1

```

となり、定理 5.4.11 より $\Lambda(A)$ の Graver 基底 $\text{Gr}_{\Lambda(A)}$ である。持ち上げにより増えた変数の部分 (y の部分) は後半の部分である。 y に 1 を代入することでその部分は消えるので、結局 A の Graver 基底は、

```

-9 10 0 -1
-8 8 1 -1
-7 6 2 -1
-6 4 3 -1
-5 2 4 -1
-4 0 5 -1
-3 -2 6 -1
-2 -4 7 -1
-1 -6 8 -1
-1 2 -1 0
0 -8 9 -1

```

に対応する 2 項式たちになる。

問題 22 例題 19 の配置行列について, Graver 基底を Lawrence 持ち上げにより計算せよ。

参考文献

- [1] J.A. De Loera, J. Rambau, F. Santos, *Triangulations: Structures for Applications and Algorithms*, 2010, Springer.
- [2] 大杉英史, 素手で計算するグレブナー基底, 数学セミナー, 2012 年 2 月号, 20–24
- [3] F. Santos, A Point Configuration whose Space of Triangulations is Disconnected, *Journal of the American Mathematical Society* **13** (2000), 611–637.
- [4] A. Schrijver, *Theory of Linear and Integer Programming*, 1986, Wiley Interscience.
- [5] B. Sturmfels, *Gröbner Bases and Convex Polytopes*, 1995, American Mathematical Society, University lecture series 8.
- [6] G. M. Ziegler, *Lectures on polytopes*, 1995, Springer,
(邦訳: 八森正泰, 岡本吉央 (訳), 凸多面体の数学, シュプリンガー・フェアラーク東京, 2003)
- [7] 4ti2 team, 4ti2—A software package for algebraic, geometric and combinatorial problems on linear spaces,
<http://www.4ti2.de/>
- [8] A.N. Jensen, Gfan, a software system for Gröbner fans and tropical varieties,
<http://www.math.tu-berlin.de/~jensen/software/gfan/gfan.html>
- [9] E. Gawrilow, M. Joswig, polymake: a Framework for Analyzing Convex Polytopes, *Polytopes — Combinatorics and Computation*, (2000), 43–74,
<http://polymake.org/>
- [10] M. Noro, et al., Risa/Asir,
<http://www.math.kobe-u.ac.jp/Asir/asir-ja.html>
- [11] J. Rambau, TOPCOM: Triangulations of Point Configurations and Oriented Matroids, *Mathematical Software—ICMS 2002*, (2002), 330–340,
<http://www.rambau.wm.uni-bayreuth.de/TOPCOM/>