Algorithms for the toric Hilbert scheme

Michael Stillman[1], Bernd Sturmfels[2] and Rekha R. Thomas[3]

[1]Cornell University, Department of Mathematics, Ithaca, NY 14853
[2]UC Berkeley, Department of Mathematics, Berkeley, CA 94720
[3]University of Washington, Department of Mathematics, Seattle, WA 98195

Abstract. The toric Hilbert scheme parametrizes all algebras isomorphic to a given semigroup algebra as a multigraded vector space. All components of the scheme are toric varieties, and among them, there is a fairly well understood coherent component. It is unknown whether toric Hilbert schemes are always connected. In this chapter we illustrate the use of *Macaulay 2* for exploring the structure of toric Hilbert schemes. In the process we will encounter algorithms from commutative algebra, algebraic geometry, polyhedral theory and geometric combinatorics.

Introduction

Consider the multigrading of the polynomial ring $R = \mathbb{C}[x_1, \ldots, x_n]$ specified by a non-negative integer $d \times n$-matrix $A = (a_1, \ldots, a_n)$ such that degree $(x_i) = a_i \in \mathbb{N}^d$. This defines a decomposition $R = \bigoplus_{b \in \mathbb{N}A} R_b$, where $\mathbb{N}A$ is the subsemigroup of $\mathbb{N}^d$ spanned by $a_1, \ldots, a_n$, and R_b is the $\mathbb{C}$-span of all monomials $x^u = x_1^{u_1} \cdots x_n^{u_n}$ with degree $Au = a_1 u_1 + \cdots + a_n u_n = b$. The *toric Hilbert scheme* $Hilb_A$ parametrizes all A-homogeneous ideals $I \subset R$ (ideals that are homogeneous under the multigrading of R by $\mathbb{N}A$) with the property that $(R/I)_b$ is a 1-dimensional $\mathbb{C}$-vectors pace, for all $b \in \mathbb{N}A$. We call such an ideal I an A-*graded* ideal. Equivalently, I is A-graded if it is A-homogeneous and R/I is isomorphic as a multigraded vector space to the semigroup algebra $\mathbb{C}[\mathbb{N}A] = R/I_A$, where

$$I_A := \langle x^u - x^v \ : \ Au = Av \rangle \subset R$$

is the *toric ideal* of A. An A-graded ideal is generated by binomials and monomials in R since, by definition, any two monomials x^u and x^v of the same degree $Au = Av$ must be $\mathbb{C}$-linearly dependent modulo the ideal.

We recommend [22, §4, §10] as an introductory reference for the topics in this chapter. The study of toric Hilbert schemes for $d = 1$ goes back to Arnold [1] and Korkina et.al.[13], and it was further developed by Sturmfels ([21] and [22, §10]). Peeva and Stillman [17] introduced the scheme structure which gives the toric Hilbert scheme its universal property, and from this they derive a formula for the tangent space of a point on $Hilb_A$. Maclagan recently showed that the quadratic binomials in [21, §5] define the same scheme as the determinantal equations in [17]. Both of these systems of global equations are generally much too big for practical computations. Instead, most of our algorithms are based on the local equations given by Peeva and Stillman in [16] and the combinatorial approach of Maclagan and Thomas in [14].

We begin with the computation of a toric ideal using *Macaulay 2*. Our running example throughout this chapter is the following 2×5-matrix:

$$A \quad = \quad \begin{pmatrix} 1\,1\,1\,1\,1 \\ 0\,1\,2\,7\,8 \end{pmatrix} \tag{1}$$

which we input to *Macaulay 2* as a list of lists of integers.

```
i1 : A = {{1,1,1,1,1},{0,1,2,7,8}};
```

The toric ideal of A lives in the multigraded ring $R := \mathbb{C}[a,b,c,d,e]$.

```
i2 : R = QQ[a..e,Degrees=>transpose A];

i3 : describe R

o3 = QQ [a, b, c, d, e, Degrees => {{1, 0}, {1, 1}, {1, 2}, {1, 7}, {1 ···

o3 : Adjacent
```

We use Algorithm 12.3 in [22] to compute I_A. The first step is to find a matrix B whose rows generate the lattice $ker_{\mathbb{Z}}(A) := \{x \in \mathbb{Z}^n : Ax = 0\}$.

```
i4 : B = transpose syz matrix A

o4 = | 1 -2 1  0 0 |
     | 0 5  -6 1 0 |
     | 0 6  -7 0 1 |

              3       5
o4 : Matrix ZZ  <--- ZZ
```

Although in theory any basis of $ker_{\mathbb{Z}}(A)$ will suffice, in practice it is more efficient to use a *reduced* basis [20, §6.2] which can be computed using the *basis reduction* package `LLL.m2` in *Macaulay 2*. The command `LLL` when applied to the output of `syz matrix A` will return a matrix of the same size whose columns form a reduced lattice basis for $ker_{\mathbb{Z}}(A)$. The output appears in compressed form as follows:

```
i5 : load "LLL.m2";

i6 : LLL syz matrix A

o6 = | .  1  2  |
     | 1  -1 .  |
     | -1 .  -3 |
     | -1 -1 2  |
     | 1  1  -1 |

              5       3
o6 : Matrix ZZ  <--- ZZ
```

We recompute B using this package to get the following 3×5 matrix.

```
i7 : B = transpose matrix LLL syz matrix A

o7 = | 0 1  -1 -1 1  |
     | 1 -1 0  -1 1  |
     | 2 0  -3 2  -1 |

              3       5
o7 : Matrix ZZ  <--- ZZ
```

The advantage of a reduced basis may not be apparent in small examples. However, as the size of A increases, it becomes increasingly important for the termination of Algorithm 12.3. (To appreciate this, consider the matrix (7) from Section 4.)

A row $b = b^+ - b^-$ of B is then coded as the binomial $x^{b^+} - x^{b^-} \in R$, and we let J be the ideal generated by all such binomials.

```
i8 : toBinomial = (b,R) -> (
          top := 1_R; bottom := 1_R;
          scan(#b, i -> if b_i > 0 then top = top * R_i^(b_i)
               else if b_i < 0 then bottom = bottom * R_i^(-b_i));
          top - bottom);

i9 : J = ideal apply(entries B, b -> toBinomial(b,R))

                                             2 2    3
o9 = ideal (- c*d + b*e, - b*d + a*e, a d  - c e)

o9 : Ideal of R
```

The toric ideal equals $(J : (x_1 \cdots x_n)^\infty)$ which is computed via n successive saturations as follows:

```
i10 : scan(gens ring J, f -> J = saturate(J,f))
```

Putting the above pieces of code together, we get the following procedure for computing the toric ideal of a matrix A.

```
i11 : toricIdeal = (A) -> (
          n := #(A_0);
          R = QQ[vars(0..n-1),Degrees=>transpose A,MonomialSize=>16];
          B := transpose matrix LLL syz matrix A;
          J := ideal apply(entries B, b -> toBinomial(b,R));
          scan(gens ring J, f -> J = saturate(J,f));
          J
          );
```

See [2], [11] and [22, §4, §12] for other algorithms for computing toric ideals and various ideas for speeding up the computation.

In our example, $I_A = \langle cd - be, bd - ae, b^2 - ac, a^2d^2 - c^3e, c^4 - a^3e, bc^3 - a^3d, ad^4 - c^2e^3, d^6 - ce^5\rangle$ which we now compute using this procedure.

```
i12 : I = toricIdeal A;

o12 : Ideal of R

i13 : transpose mingens I

o13 = {-2, -9}  | cd-be     |
      {-2, -8}  | bd-ae     |
      {-2, -2}  | b2-ac     |
      {-4, -14} | a2d2-c3e  |
      {-4, -8}  | c4-a3e    |
      {-4, -7}  | bc3-a3d   |
      {-5, -28} | ad4-c2e3  |
      {-6, -42} | d6-ce5    |

               8       1
o13 : Matrix R  <--- R
```

This ideal defines an embedding of $\mathbb{P}^1$ as a degree 8 curve into $\mathbb{P}^4$. We will see in Section 3 that its toric Hilbert scheme $Hilb_A$ has a non-reduced component.

This chapter is organized into four sections and two appendices as follows. The main goal in Section 1 is to describe an algorithm for generating all monomial A-graded ideals for a given A. These monomial ideals are the vertices of the *flip graph* of A whose connectivity is equivalent to the connectivity of $Hilb_A$. We describe how all neighbors of a given vertex of this graph can be calculated. In Section 2, we explain the role of polyhedral geometry in the study of $Hilb_A$. Our first algorithm tests for *coherence* in a monomial A-graded ideal. We then show how to compute the polyhedral complexes supporting A-graded ideals which in turn relates the flip graph of A to the *Baues graph* of A. For unimodular matrices, these two graphs coincide and hence our method of computing the flip graph can be used to compute the Baues graph. Section 3 explores the components of $Hilb_A$ via local equations around the torus fixed points of the scheme. We include a combinatorial interpretation of these local equations from the point of view of integer programming. The scheme $Hilb_A$ has a *coherent* component which is examined in detail in Section 4. We prove that this component is, in general, not normal and that its normalization is the toric variety of the Gröbner fan of I_A. We conclude the chapter with two appendices, each containing one large piece of *Macaulay 2* code that we use in this chapter. Appendix 1 contains the *Macaulay 2* file `polarCone.m2` which is used to convert a generator representation of a polyhedron to an inequality representation and vice versa. Appendix 2 contains `minPres.m2` for computing minimal presentations of polynomial quotient rings. The main ingredient of this package is the subroutine `removeRedundantVariables` which is what we use in this chapter.

1 Generating monomial ideals

We start out by computing the *Graver basis* Gr_A, which is the set of binomials in I_A that are minimal with respect to the partial order

$$x^u - x^v \leq x^{u'} - x^{v'} \quad :\Longleftrightarrow \quad x^u \text{ divides } x^{u'} \text{ and } x^v \text{ divides } x^{v'}.$$

The set Gr_A is a *universal Gröbner basis* of I_A and has its origins in the theory of integer programming [9]. It can be computed using [22, Algorithm 7.2], a *Macaulay 2* version of which is given below.

```
i14 : graver = (I) -> (
          R := ring I;
          k := coefficientRing R;
          n := numgens R;
          -- construct new ring S with 2n variables
          S := k[Variables=>2*n,MonomialSize=>16];
          toS := map(S,R,(vars S)_{0..n-1});
          toR := map(R,S,vars R | matrix(R, {toList(n:1)}));
          -- embed I in S
```

```
 m := gens toS I;
 -- construct the toric ideal of the Lawrence
 -- lifting of A
 i := 0;
 while i < n do (
      wts := join(toList(i:0),{1},toList(n-i-1:0));
      wts = join(wts,wts);
      m = homogenize(m,S_(n+i),wts);
      i=i+1;
      );
J := ideal m;
scan(gens ring J, f -> J = saturate(J,f));
-- apply the map toR to the minimal generators of J
f := matrix entries toR mingens J;
p := sortColumns f;
f_p) ;
```

The above piece of code first constructs a new polynomial ring S in n more variables than R. Assume $S = \mathbb{C}[x_1, \ldots, x_n, y_1, \ldots, y_n]$. The inclusion map `toS` $: R \to S$ embeds the toric ideal I in S and collects its generators in the matrix `m`. A binomial $x^a - x^b$ lies in Gr_A if and only if $x^a y^b - x^b y^a$ is a minimal generator of the toric ideal in S of the $(d+n) \times 2n$ matrix

$$\Lambda(A) := \begin{pmatrix} A & 0 \\ I_n & I_n \end{pmatrix}$$

which is called the *Lawrence lifting* of A. Since $u \in ker_{\mathbb{Z}}(A) \Leftrightarrow (u, -u) \in ker_{\mathbb{Z}}(\Lambda(A))$, we use the `while loop` to homogenize the binomials in `m` with respect to $\Lambda(A)$, using the n new variables in S. This converts a binomial $x^a - x^b \in$ `m` to the binomial $x^a y^b - x^b y^a$. The ideal generated by these new binomials is labeled J. As before, we can now successively saturate J to get the toric ideal of $\Lambda(A)$ in S. The image of the minimal generators of this toric ideal under the map `toR` $: S \to R$ such that $x_i \mapsto x_i$ and $y_i \mapsto 1$ is precisely the Graver basis Gr_A. These binomials are the entries of the matrix `f` and is output by the program.

In our example Gr_A consists of 42 binomials.

```
i15 : Graver = graver I

o15 = {0, 0} | -cd+be -bd+ae -b2+ac -cd2+ae2 -a2d2+c3e -c4+a2bd -c4+a3 ···

                1        42
o15 : Matrix R  <--- R
```

Returning to the general case, an element b of $\mathbb{N}A$ is called a *Graver degree* if there exists a binomial $x^u - x^v$ in the Graver basis Gr_A such that $Au = Av = b$. If b is a Graver degree then the set of monomials in R_b is the corresponding *Graver fiber*. In our running example there are 37 distinct Graver fibers. We define the `ProductIdeal` of A as $PI := \langle x^a x^b : x^a - x^b \in Gr_A \rangle$. This ideal is contained in every monomial ideal of $Hilb_A$ and hence no monomial in PI can be a standard monomial of a monomial A-graded ideal. Since our purpose in constructing Graver fibers is to use them to generate all monomial A-graded ideals, we will be content with listing just the monomials in each Graver fiber that do not lie in PI. Since R is multigraded by A, we can obtain such a presentation of a Graver fiber by simply asking for the basis of R in degree b modulo PI.

```
i16 : graverFibers = (Graver) -> (
          ProductIdeal := (I) -> ( trim ideal(
             apply(numgens I, a -> (
                 f := I_a; leadTerm f * (leadTerm f - f)))));
          PI := ProductIdeal ideal Graver;
          R := ring Graver;
          new HashTable from apply(
              unique degrees source Graver,
              d -> d => compress (basis(d,R) % PI) ));

i17 : fibers = graverFibers Graver

o17 = HashTable{{2, 2} => {0, 0} | ac b2 |                              ...
                {2, 8} => {0, 0} | ae bd |
                {2, 9} => {0, 0} | be cd |
                {3, 16} => {0, 0} | ae2 bde cd2 |
                {4, 14} => {0, 0} | a2d2 c3e |
                {4, 7} => {0, 0} | a3d bc3 |
                {4, 8} => {0, 0} | a3e a2bd c4 |
                {5, 10} => {0, 0} | a3ce a2b2e a2bcd ab3d c5 |
                {5, 14} => {0, 0} | a3d2 ac3e b2c2e bc3d |
                {5, 16} => {0, 0} | a3e2 a2cd2 ab2d2 c4e |
                {5, 21} => {0, 0} | a2d3 bc2e2 c3de |
                {5, 22} => {0, 0} | a2d2e abd3 c3e2 |
                {5, 28} => {0, 0} | ad4 c2e3 |
                {5, 7} => {0, 0} | a4d abc3 b3c2 |
                {5, 8} => {0, 0} | a4e a3bd ac4 b2c3 |
                {6, 12} => {0, 0} | a3c2e a2bc2d ab4e b5d c6 |
                {6, 14} => {0, 0} | a4d2 a2c3e abc3d b4ce b3c2d |
                {6, 18} => {0, 0} | a3ce2 a2b2e2 a2c2d2 b4d2 c5e |
                {6, 21} => {0, 0} | a3d3 abc2e2 ac3de b3ce2 bc3d2 |
                {6, 24} => {0, 0} | a3e3 a2cd2e abcd3 b3d3 c4e2 |
                {6, 28} => {0, 0} | a2d4 ac2e3 b2ce3 c3d2e |
                {6, 30} => {0, 0} | a2d2e2 acd4 b2d4 c3e3 |
                {6, 35} => {0, 0} | ad5 bce4 c2de3 |
                {6, 36} => {0, 0} | ad4e bd5 c2e4 |
                {6, 42} => {0, 0} | ce5 d6 |
                {6, 7} => {0, 0} | a5d a2bc3 b5c |
                {6, 8} => {0, 0} | a5e a4bd a2c4 b4c2 |
                {7, 14} => {0, 0} | a5d2 a3c3e a2bc3d b6e b5cd c7 |
                {7, 21} => {0, 0} | a4d3 a2bc2e2 a2c3de abc3d2 b5e2 b3 ...
                {7, 28} => {0, 0} | a3d4 a2c2e3 ac3d2e b4e3 bc3d3 |
                {7, 35} => {0, 0} | a2d5 abce4 ac2de3 b3e4 c3d3e |
                {7, 42} => {0, 0} | ace5 ad6 b2e5 c2d2e3 |
                {7, 49} => {0, 0} | be6 cde5 d7 |
                {7, 7} => {0, 0} | a6d a3bc3 b7 |
                {7, 8} => {0, 0} | a6e a5bd a3c4 b6c |
                {8, 56} => {0, 0} | ae7 bde6 cd2e5 d8 |
                {8, 8} => {0, 0} | a7e a6bd a4c4 b8 |

o17 : HashTable
```

For example, the Graver degree $(8, 8)$ corresponds to the Graver fiber

$$\{\,\underline{a^7e},\,\underline{a^6bd},\,\underline{a^4c^4},\,a^3b^2c^3,\,a^2b^4c^2,\,ab^6c,\,\underline{b^8}\,\}.$$

Our *Macaulay 2* code only outputs the four underlined monomials, in the format `| a7e a6bd a4c4 b8 |`. The three non-underlined monomials lie in the `ProductIdeal`. Graver degrees are important because of the following result.

Lemma 1. *[22, Lemma 10.5]. The multidegree of any minimal generator of any ideal I in $Hilb_A$ is a Graver degree.*

The next step in constructing the toric Hilbert scheme is to compute all its fixed points with respect to the scaling action of the n-dimensional algebraic torus $(\mathbb{C}^*)^n$. (The torus $(\mathbb{C}^*)^n$ acts on R by scaling variables : $\lambda \mapsto \lambda \cdot x := (\lambda_1 x_1, \ldots, \lambda_n x_n)$.) These fixed points are the monomial ideals M lying on $Hilb_A$. Every term order $\prec$ on the polynomial ring R gives such a monomial ideal: $M = in_\prec(I_A)$, the initial ideal of the toric ideal I_A with respect to $\prec$. Two ideals J and J' are said to be *torus isomorphic* if $J = \lambda \cdot J'$ for some $\lambda \in (\mathbb{C}^*)^n$. Any monomial A-graded ideal that is torus isomorphic to an initial ideal of I_A is said to be *coherent*. In particular, the initial ideals of I_A are coherent and they can be computed by [22, Algorithm 3.6] applied to I_A. A refinement and fast implementation can be found in the software package `TiGERS` by Huber and Thomas [12].

Now we wish to compute all monomial ideals M on $Hilb_A$ regardless of whether M is coherent or not. For this we use the procedure `generateAmonos` given below. This procedure takes in the Graver basis Gr_A and records the numerator of the Hilbert series of I_A in `trueHS`. It then computes the Graver fibers of A, sorts them and calls the subroutine `selectStandard` to generate a candidate for a monomial ideal on $Hilb_A$.

```
i18 : generateAmonos = (Graver) -> (
           trueHS := poincare coker Graver;
           fibers := graverFibers Graver;
           fibers = apply(sort pairs fibers, last);
           monos = {};
           selectStandard := (fibers, J) -> (
           if #fibers == 0 then (
              if trueHS == poincare coker gens J
              then (monos = append(monos,flatten entries mingens J));
           ) else (
              P := fibers_0;
              fibers = drop(fibers,1);
              P = compress(P % J);
              nP := numgens source P;
              -- nP is the number of monomials not in J.
              if nP > 0 then (
                 if nP == 1 then selectStandard(fibers,J)
                 else (--remove one monomial from P,take the rest.
                       P = flatten entries P;
                       scan(#P, i -> (
                            J1 := J + ideal drop(P,{i,i});
                            selectStandard(fibers, J1)))));
           ));
           selectStandard(fibers, ideal(0_(ring Graver)));
           ) ;
```

The input to the subroutine `selectStandard` are the Graver fibers given as a list of matrices and a monomial ideal J which should be included in every A-graded ideal that we generate. The subroutine then loops through each Graver fiber, and at each step selects a standard monomial from that fiber and updates the ideal J by adding the other monomials in this fiber to J. The final J output by the subroutine is the candidate ideal that is sent back to `generateAmonos`. It is stored by the program if its Hilbert series agrees with that of I_A. All the monomial A-graded ideals are stored in the list `monos`. Below, we ask *Macaulay 2* for the cardinality of `monos` and its first ten elements.

```
i19 : generateAmonos Graver;

i20 : #monos

o20 = 281

i21 : scan(0..9, i -> print toString monos#i)
{c*d, b*d, b^2, c^3*e, c^4, b*c^3, c^2*e^3, b*c^2*e^2, b*c*e^4, d^6}
{c*d, b*d, b^2, c^3*e, c^4, b*c^3, c^2*e^3, b*c^2*e^2, c*e^5, b*c*e^4, ···
{c*d, b*d, b^2, c^3*e, c^4, b*c^3, c^2*e^3, b*c^2*e^2, c*e^5, b*c*e^4, ···
{c*d, b*d, b^2, c^3*e, c^4, b*c^3, c^2*e^3, b*c^2*e^2, c*e^5, b*c*e^4, ···
{c*d, b*d, b^2, c^3*e, c^4, b*c^3, c^2*e^3, b*c^2*e^2, d^6, a*d^5}
{c*d, b*d, b^2, c^3*e, c^4, b*c^3, b*c^2*e^2, a*d^4, d^6}
{c*d, b*d, b^2, c^3*e, c^4, b*c^3, a*d^4, a^2*d^3, d^6}
{c*d, b*d, b^2, a^2*d^2, c^4, b*c^3, a*d^4, d^6}
{c*d, b*d, b^2, a^2*d^2, a^3*d, c^4, a*d^4, d^6}
{c*d, b*d, b^2, a^3*e, a^2*d^2, a^3*d, a*d^4, d^6}
```

The monomial ideals (torus-fixed points) on $Hilb_A$ form the vertices of the *flip graph* of A whose edges correspond to the torus-fixed curves on $Hilb_A$. This graph was introduced in [14] and provides structural information about $Hilb_A$. The edges emanating from a monomial ideal M can be constructed as follows: For any minimal generator x^u of M, let x^v be the unique monomial with $x^v \notin M$ and $Au = Av$. Form the *wall ideal* which is generated by $x^u - x^v$ and all minimal generators of M other than x^u, and let M' be the initial monomial ideal of the wall ideal with respect to any term order $\succ$ for which $x^v \succ x^u$. It can be shown that M' is the unique initial monomial ideal of the wall ideal that contains x^v. If M' lies on $Hilb_A$ then $\{M, M'\}$ is an edge of the flip graph. We now illustrate the *Macaulay 2* procedure for computing all flip neighbors of a monomial A-graded ideal.

```
i22 : findPositiveVector = (m,s) -> (
           expvector := first exponents s - first exponents m;
           n := #expvector;
           i := first positions(0..n-1, j -> expvector_j > 0);
           splice {i:0, 1, (n-i-1):0}
           );

i23 : flips = (M) -> (
           R := ring M;
          -- store generators of M in monoms
           monoms := first entries generators M;
           result := {};
          -- test each generator of M to see if it leads to a neighbor ···
           scan(#monoms, i -> (
             m := monoms_i;
             rest := drop(monoms,{i,i});
             b := basis(degree m, R);
             s := (compress (b % M))_(0,0);
             J := ideal(m-s) + ideal rest;
             if poincare coker gens J == poincare coker gens M then (
               w := findPositiveVector(m,s);
               R1 := (coefficientRing R)[generators R, Weights=>w];
               J = substitute(J,R1);
               J = trim ideal leadTerm J;
               result = append(result,J);
                )));
           result
      );
```

The above code inputs a monomial A-graded ideal M whose minimal generators are stored in the list `monoms`. The flip neighbors of M will be stored in `result`. For each monomial x^u in `monoms` we need to test whether it yields a flip neighbor of M or not. At the i-th step of this loop, we let `m` be the i-th monomial in `monoms`. The list `rest` contains all monomials in `monoms` except `m`. We compute the standard monomial `s` of M of the same degree as m. The wall ideal of $m - s$ is the binomial ideal J generated by $m - s$ and the monomials in `rest`. We then check whether J is A-graded by comparing its Hilbert series with that of M. (Alternately, one could check whether M is the initial ideal of the wall ideal with respect to $m \succ s$.) If this is the case, we use the subroutine `findPositiveVector` to find a unit vector $w = (0, \ldots, 1, \ldots, 0)$ such that $w \cdot s > w \cdot m$. The flip neighbor is then the initial ideal of J with respect to w and it is stored in `result`. The program outputs the minimal generators of each flip neighbor. Here is an example.

```
i24 : R = QQ[a..e,Degrees=>transpose A];

i25 : M = ideal(a*e,c*d,a*c,a^2*d^2,a^2*b*d,a^3*d,c^2*e^3,
               c^3*e^2,c^4*e,c^5,c*e^5,a*d^5,b*e^6);

o25 : Ideal of R

i26 : F = flips M

                                    2 2    3    4   2 3   3 2     5     5     ···
o26 = {ideal (a*e, c*d, a*c, a d , a d, c , c e , c e , a*d , c*e , b* ···

o26 : List

i27 : #F

o27 = 4

i28 : scan(#F, i -> print toString entries mingens F_i)
{{a*e, c*d, a*c, a^2*d^2, a^3*d, c^4, c^2*e^3, c^3*e^2, a*d^5, c*e^5,  ···
{{c*d, a*e, a*c, a^2*d^2, a^2*b*d, a^3*d, c^3*e^2, c^4*e, c^5, a*d^4,  ···
{{a*e, c*d, a*c, a^2*d^2, a^3*d, a^2*b*d, c^2*e^3, c^3*e^2, c^4*e, c^5 ···
{{a*e, a*c, c*d, a^2*b*d, a^3*d, a^2*d^2, c^2*e^3, c^3*e^2, c^4*e, c^5 ···
```

It is an open problem whether the toric Hilbert scheme $Hilb_A$ is connected. Recent work in geometric combinatorics [19] suggests that this is probably false for some A. This result and its implications for $Hilb_A$ will be discussed further in Section 2. The following theorem of Maclagan and Thomas [14] reduces the connectivity of $Hilb_A$ to a combinatorial problem.

Theorem 2. *The toric Hilbert scheme $Hilb_A$ is connected if and only if the flip graph of A is connected.*

We now have two algorithms for listing monomial ideals on $Hilb_A$. First, there is the *backtracking algorithm* whose *Macaulay 2* implementation was described above. Second, there is the *flip search algorithm* which starts with any coherent monomial ideal M and then constructs the connected component of M in the flip graph of A by carrying out local flips as above. This procedure is also implemented in `TiGERS` [12]. Clearly, the two algorithms will produce the same answer if and only if $Hilb_A$ is connected. In other words, finding an example where $Hilb_A$ is disconnected is equivalent to finding a matrix A for which the flip search algorithm produces fewer monomial ideals than the backtracking algorithm.

2 Polyhedral Geometry

Algorithms from polyhedral geometry are essential in the study of the toric Hilbert scheme. Consider the problem of deciding whether or not a given monomial ideal M in $Hilb_A$ is coherent. This problem gives rise to a system of linear inequalities as follows: Let $x^{u_1}, \ldots, x^{u_r}$ be the minimal generators of M, and let x^{v_i} be the unique standard monomial with $Au_i = Av_i$. Then M is coherent if and only if there exists a vector $w \in \mathbb{R}^n$ such that $w \cdot (u_i - v_i) > 0$ for $i = 1, \ldots, r$. Thus the test for coherence amounts to solving a *feasibility problem of linear programming*, and there are many highly efficient algorithms (based on the simplex algorithms or interior point methods) available for this task. For our experimental purposes, it is convenient to use the code `polarCone.m2`, given in Appendix 1, which is based on the (inefficient but easy-to-implement) *Fourier-Motzkin elimination* method (see [25] for a description). This code converts the generator representation of a polyhedron to its inequality representation and vice versa. A simple example is given in Appendix 1. In particular, given a Gröbner basis $\mathcal{G}$ of I_A, the function `polarCone` will compute all the extreme rays of the *Gröbner cone* $\{w \in \mathbb{R}^n \ : \ w \cdot (u_i - v_i) \geq 0 \text{ for each } x^{u_i} - x^{v_i} \in \mathcal{G}\}$.

We now show how to use *Macaulay 2* to decide whether a monomial A-graded ideal M is coherent. The first step in this calculation is to compute all the standard monomials of M of the same degree as the minimal generators of M. We do this using the procedure `stdMonomials`.

```
i29 : stdMonomials = (M) -> (
          R := ring M;
          RM := R/M;
          apply(numgens M, i -> (
                s := basis(degree(M_i),RM); lift(s_(0,0), R)))
          );
```

As an example, consider the following monomial A-graded ideal.

```
i30 : R = QQ[a..e,Degrees => transpose A ];

i31 : M = ideal(a^3*d, a^2*b*d, a^2*d^2, a*b^3*d, a*b^2*d^2, a*b*d^3,
                a*c, a*d^4, a*e, b^5*d, b^4*d^2, b^3*d^3, b^2*d^4,
                b*d^5, b*e, c*e^5);

o31 : Ideal of R

i32 : toString stdMonomials M

o32 = {b*c^3, c^4, c^3*e, c^5, c^4*e, c^3*e^2, b^2, c^2*e^3, b*d, c^6, ···

o32 : String
```

From the pairs x^u, x^v of minimal generators x^u and corresponding standard monomials x^v, the function `inequalities` creates a matrix whose columns are the vectors $u - v$.

```
i33 : inequalities = (M) -> (
              stds := stdMonomials(M);
              transpose matrix apply(numgens M, i -> (
                  flatten exponents(M_i) -
                      flatten exponents(stds_i))));

i34 : inequalities M

o34 = | 3  2  2  1  1  1  1  1  1  0  0  0  0  0  0  0  |
      | -1 1  0  3  2  1  -2 0  -1 5  4  3  2  1  1  0  |
      | -3 -4 -3 -5 -4 -3 1  -2 0  -6 -5 -4 -3 -2 -1 1  |
      | 1  1  2  1  2  3  0  4  -1 1  2  3  4  5  -1 -6 |
      | 0  0  -1 0  -1 -2 0  -3 1  0  -1 -2 -3 -4 1  5  |

               5         16
o34 : Matrix ZZ  <--- ZZ
```

It is convenient to simplify the output of the next procedure using the following program to divide an integer vector by the g.c.d. of its components. We also load `polarCone.m2` which is needed in `decideCoherence` below.

```
i35 : primitive := (L) -> (
           n := #L-1; g := L#n;
           while n > 0 do (n = n-1; g = gcd(g, L#n););
           if g === 1 then L else apply(L, i -> i // g));

i36 : load "polarCone.m2"

i37 : decideCoherence = (M) -> (
           ineqs := inequalities M;
           c := first polarCone ineqs;
           m := - sum(numgens source c, i -> c_{i});
           prods := (transpose m) * ineqs;
           if numgens source prods != numgens source compress prods
           then false else primitive (first entries transpose m));
```

Let K be the cone $\{x \in \mathbb{R}^n : g \cdot x \leq 0,$ for all columns g of `ineqs` $\}$. The command `polarCone ineqs` computes a pair of matrices P and Q such that K is the sum of the cone generated by the columns of P and the subspace generated by the columns of Q. Let `m` be the negative of the sum of the columns of P. Then `m` lies in the cone $-K$. The entries in the matrix `prods` are the dot products $g \cdot m$ for each column g of `ineqs`. Since M is a monomial A-graded ideal, it is coherent if and only if K is full dimensional which is the case if and only if no dot product $g \cdot m$ is zero. This is the conditional in the `if .. then` statement of `decideCoherence`. If M is coherent, the program outputs the primitive representative of `m` and otherwise returns the boolean `false`. Notice that if M is coherent, the cone $-K$ is the Gröbner cone corresponding to M and the vector `m` is a weight vector w such that $in_w(I_A) = M$. We now test whether the ideal M from line `i29` is coherent.

```
i38 : decideCoherence(M)

o38 = {0, 0, 1, 15, 18}

o38 : List
```

Hence, M is coherent: it is the initial ideal with respect to the weight vector $w = (0, 0, 1, 15, 18)$ of the toric ideal in our running example (1). Here is one of the 55 noncoherent monomial A-graded ideals of this matrix.

```
i39 : N = ideal(a*e,c*d,a*c,c^3*e,a^3*d,c^4,a*d^4,a^2*d^3,c*e^5,
                c^2*e^4,d^7);

o39 : Ideal of R

i40 : decideCoherence(N)

o40 = false
```

In the rest of this section, we study the connection between A-graded ideals and polyhedral complexes defined on A which will relate the flip graph to the Baues graph of A. (See [18] for a survey of the Baues problem and its relatives). Let $pos(A) := \{Au : u \in \mathbb{R}^n, u \geq 0\}$ be the cone generated by the columns of A in $\mathbb{R}^d$. A *polyhedral subdivision* Δ of A is a collection of full dimensional subcones $pos(A_\sigma)$ of $pos(A)$ such that the union of these subcones is $pos(A)$ and the intersection of any two subcones is a face of each. Here $A_\sigma := \{a_j : j \in \sigma \subseteq \{1, \dots, n\}\}$. It is customary to identify Δ with the set of sets $\{\sigma : pos(A_\sigma) \in \Delta\}$. If every cone in the subdivision Δ is simplicial (the number of extreme rays of the cone equals the dimension of the cone), we say that Δ is a *triangulation* of A. The simplicial complex corresponding to a triangulation Δ is uniquely obtained by including in Δ all the subsets of every $\sigma \in \Delta$. We refer the reader to [22, §8] for more details.

For each $\sigma \in \Delta$, let I_σ be the prime ideal that is the sum of the toric ideal I_{A_σ} and the monomial ideal $\langle x_j : j \notin \sigma \rangle$. Recall that two ideals J and J' are said to be *torus isomorphic* if $J = \lambda \cdot J'$ for some $\lambda \in (\mathbb{C}^*)^n$. The following theorem shows that polyhedral subdivisions of A are related to A-graded ideals via their radicals.

Theorem 3. *(Theorem 10.10 [22, §10]) If I is any A-graded ideal, then there exists a polyhedral subdivision $\Delta(I)$ of A such that $\sqrt{I} = \cap_{\sigma \in \Delta(I)} J_\sigma$ where each component J_σ is a prime ideal that is torus isomorphic to I_σ.*

We say that $\Delta(I)$ supports the A-graded ideal I. When M is a monomial A-graded ideal, $\Delta(M)$ is a triangulation of A. In particular, if M is coherent (i.e, $M = in_w(I_A)$ for some weight vector w), then $\Delta(M)$ is the *regular* or *coherent* triangulation of A induced by w [22, §8]. The coherent triangulations of A are in bijection with the vertices of the *secondary polytope* of A [3], [8].

It is convenient to represent a triangulation Δ of A by its *Stanley-Reisner* ideal $I_\Delta := \langle x_{i_1} x_{i_2} \cdots x_{i_k} : \{i_1, i_2, \dots, i_k\}$ is a non-face of $\Delta\rangle$. If M is a monomial A-graded ideal, Theorem 3 implies that $I_{\Delta(M)}$ is the radical of M. Hence we will represent triangulations of A by their Stanley-Reisner ideals. As seen below, the matrix in our running example has eight distinct triangulations corresponding to the eight distinct radicals of the 281 monomial A-graded ideals computed earlier. All eight are coherent.

$$\begin{array}{lcl}
\{\{1,2\},\{2,3\},\{3,4\},\{4,5\}\} & \leftrightarrow & \langle ac, ad, ae, bd, be, ce\rangle \\
\{\{1,3\},\{3,4\},\{4,5\}\} & \leftrightarrow & \langle b, ad, ae, ce\rangle \\
\{\{1,2\},\{2,4\},\{4,5\}\} & \leftrightarrow & \langle c, ad, ae, be\rangle \\
\{\{1,2\},\{2,3\},\{3,5\}\} & \leftrightarrow & \langle d, ac, ae, be\rangle \\
\{\{1,3\},\{3,5\}\} & \leftrightarrow & \langle b, d, ae\rangle \\
\{\{1,4\},\{4,5\}\} & \leftrightarrow & \langle b, c, ae\rangle \\
\{\{1,2\},\{2,5\}\} & \leftrightarrow & \langle c, d, ae\rangle \\
\{\{1,5\}\} & \leftrightarrow & \langle b, c, d\rangle
\end{array}$$

The Baues graph of A is a graph on all the triangulations of A in which two triangulations are adjacent if they differ by a single *bistellar flip* [18]. The *Baues problem* from discrete geometry asked whether the Baues graph of a point configuration can be disconnected for some A. Every edge of the secondary polytope of A corresponds to a bistellar flip and hence, the subgraph of the Baues graph that is induced by the coherent triangulations of A is indeed connected: it is precisely the edge graph of the secondary polytope of A. The Baues problem was recently settled by Santos [19] who gave an example of a six dimensional point configuration with 324 points for which there is an isolated (necessarily non-regular) triangulation.

Santos' configuration would also have a disconnected flip graph and hence a disconnected toric Hilbert scheme if it were true that *every* triangulation of A supports a monomial A-graded ideal. However, Peeva has shown that this need not be the case (Theorem 10.13 in [22, §10]). Hence, the map from the set of all monomial A-graded ideals to the set of all triangulations of A that sends $M \mapsto \Delta(M)$ is not always surjective, and it is unknown whether Santos' 6×324 configuration has a disconnected toric Hilbert scheme.

Thus, even though one cannot in general conclude that the existence of a disconnected Baues graph implies the existence of a disconnected flip graph, there is an important special situation in which such a conclusion is possible. We call an integer matrix A of full row rank *unimodular* if the absolute value of each of its non-zero maximal minors is the same constant. A matrix A is unimodular if and only if every monomial A-graded ideal is square-free. For a unimodular matrix A, the Baues graph of A coincides with the flip graph of A. As you might expect, Santos' configuration is not unimodular.

Theorem 4. *(Lemma 10.14 [22, §10]) If A is unimodular, then each triangulation of A supports a unique (square-free) monomial A-graded ideal. In this case, a monomial A-graded ideal is coherent if and only if the triangulation supporting it is coherent.*

Using Theorem 4 we can compute all the triangulations of a unimodular matrix since they are precisely the polyhedral complexes supporting monomial A-graded ideals. Then we could enumerate the connected component of a coherent monomial A-graded ideal in the flip graph of A to decide whether the Baues/flip graph is disconnected.

Let Δ_r be the standard r-simplex which is the convex hull of the $r+1$ unit vectors in $\mathbb{R}^{r+1}$ and let $A(r,s)$ be the $(r+s+2) \times (r+1)(s+1)$ matrix whose columns are the products of the vertices of Δ_r and Δ_s. All matrices of type $A(r,s)$ are unimodular. From the product of two triangles we get

$$A(2,2) := \begin{pmatrix} 1\,1\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,1\,1 \\ 1\,0\,0\,1\,0\,0\,1\,0\,0 \\ 0\,1\,0\,0\,1\,0\,0\,1\,0 \\ 0\,0\,1\,0\,0\,1\,0\,0\,1 \end{pmatrix}.$$

We can now use our algebraic algorithms to compute all the triangulations of $A(2,2)$. Since *Macaulay 2* requires the first entry of the degree of every variable in a ring to be positive, we use the following matrix with the same row space as $A(2,2)$ for our computation:

```
i41 : A22 =
        {{1,1,1,1,1,1,1,1,1},{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}};

i42 : I22 = toricIdeal A22

o42 = ideal (f*h - e*i, c*h - b*i, f*g - d*i, e*g - d*h, c*g - a*i, b* ···

o42 : Ideal of R
```

The ideal `I22` is generated by the 2 by 2 minors of a 3 by 3 matrix of indeterminates. This is the ideal of $\mathbb{P}^2 \times \mathbb{P}^2$ embedded in $\mathbb{P}^8$ via the Segre embedding.

```
i43 : Graver22 = graver I22;

               1       15
o43 : Matrix R  <--- R

i44 : generateAmonos(Graver22);

i45 : #monos

o45 = 108

i46 : scan(0..9,i->print toString monos#i)
{f*h, c*h, f*g, e*g, c*g, b*g, c*e, c*d, b*d}
{f*h, d*h, c*h, f*g, c*g, b*g, c*e, c*d, b*d}
{d*i, f*h, d*h, c*h, c*g, b*g, c*e, c*d, b*d}
{e*i, c*h, f*g, e*g, c*g, b*g, c*e, c*d, b*d}
{e*i, d*i, c*h, e*g, c*g, b*g, c*e, c*d, b*d}
{e*i, d*i, d*h, c*h, c*g, b*g, c*e, c*d, b*d}
{f*h, c*h, f*g, e*g, c*g, b*g, c*e, a*e, c*d}
{e*i, c*h, f*g, e*g, c*g, b*g, c*e, a*e, c*d, b*d*i}
{e*i, c*h, f*g, e*g, c*g, b*g, c*e, a*e, c*d, a*f*h}
{e*i, d*i, c*h, e*g, c*g, b*g, c*e, a*e, c*d}
```

Thus there are 108 monomial $A(2,2)$-graded ideals and `decideCoherence` will check that all of them are coherent. Since $A(2,2)$ is unimodular, each monomial $A(2,2)$-graded ideal is square-free and is hence radical. These 108 ideals represent the 108 triangulations of $A(2,2)$ and we have listed ten of them above. The flip graph (equivalently, Baues graph) of $A(2,2)$ is connected. However, it is unknown whether the Baues graph of $A(r,s)$ is connected for all values of (r,s).

3 Local Equations

Consider the reduced Gröbner basis of a toric ideal I_A for a term order w:

$$\{\, x^{u_1} - x^{v_1}\,,\; x^{u_2} - x^{v_2}\,,\; \ldots\,,\; x^{u_r} - x^{v_r} \,\}. \tag{2}$$

The initial ideal $M = in_w(I_A) = \langle x^{u_1}, x^{u_2}, \ldots, x^{u_r}\rangle$ is a coherent monomial A-graded ideal. In particular, it is a $(\mathbb{C}^*)^n$-fixed point on the toric Hilbert scheme $Hilb_A$. We shall explain a method, due to Peeva and Stillman [16], for computing local equations of $Hilb_A$ around such a fixed point. A variant of this method also works for computing the local equations around a non-coherent monomial ideal M, but that variant involves local algebra, specifically Mora's tangent cone algorithm, which is not yet fully implemented in *Macaulay 2*. See [16] for details.

We saw how to compute the flip graph of A in Section 1. The vertices of this graph are the $(\mathbb{C}^*)^n$-fixed points M and its edges correspond to the $(\mathbb{C}^*)^n$-fixed curves. By computing and decomposing the local equations around each M, we get a complete description of the scheme $Hilb_A$.

The first step is to introduce a new variable z_i for each binomial in our Gröbner basis (2) and to consider the following r binomials:

$$x^{u_1} - z_1 \cdot x^{v_1}\,,\; x^{u_2} - z_2 \cdot x^{v_2}\,,\; \ldots\,,\; x^{u_r} - z_r \cdot x^{v_r} \tag{3}$$

in the polynomial ring $\mathbb{C}[x,z]$ in $n+r$ indeterminates. The term order w can be extended to an elimination term order in $\mathbb{C}[x,z]$ so that x^{u_i} is the leading term of $x^{u_i} - z_i \cdot x^{v_i}$ for all i. We compute the minimal first syzygies of the monomial ideal M, and form the corresponding S-pairs of binomials in (3). For each S-pair

$$\frac{lcm(x^{u_i}, x^{u_j})}{x^{u_i}} \cdot (x^{u_i} - z_i \cdot x^{v_i}) \;-\; \frac{lcm(x^{u_i}, x^{u_j})}{x^{u_j}} \cdot (x^{u_j} - z_j \cdot x^{v_j})$$

we compute a normal form with respect to (3) using the extended term order w. The result is a binomial in $\mathbb{C}[x,z]$ which factors as

$$x^{\alpha} \cdot z^{\beta} \cdot (z^{\gamma} - z^{\delta}),$$

where $\alpha \in \mathbb{N}^n$ and $\beta, \gamma, \delta \in \mathbb{N}^r$. Note that this normal form is not unique but depends on our choice of a reduction path. Let J_M denote the ideal in $\mathbb{C}[z_1, \ldots, z_r]$ generated by all binomials $z^{\beta} \cdot (z^{\gamma} - z^{\delta})$ gotten from normal forms of all the S-pairs considered above.

Proposition 5. *[16] The ideal J_M is independent of the reduction paths chosen. It defines a subscheme of $\mathbb{C}^r$ isomorphic to an affine open neighborhood of the point M on the toric Hilbert scheme $Hilb_A$.*

We apply this technique to compute a particularly interesting affine chart of $Hilb_A$ for our running example. Consider the following set of 13 binomials:

$$\begin{gathered}\{\, ae - z_1 bd,\ cd - z_2 be,\ ac - z_3 b^2,\ a^2d^2 - z_4 c^3 e,\ a^2bd - z_5 c^4, \\ a^3 d - z_6 bc^3,\ c^2e^3 - z_7 ad^4,\ c^3e^2 - z_8 abd^3,\ c^4 e - z_9 ab^2d^2, \\ c^5 - z_{10} ab^3 d,\ ce^5 - z_{11} d^6,\ ad^5 - z_{12} bce^4,\ be^6 - z_{13} d^7 \,\}.\end{gathered}$$

If we set $z_1 = z_2 = \cdots = z_{13} = 1$ then we get a generating set for the toric ideal I_A. The 13 monomials obtained by setting $z_1 = z_2 = \cdots = z_{13} = 0$ generate the initial monomial ideal $M = in_w(I_A)$ with respect to the weight vector $w = (9, 3, 5, 0, 0)$. Thus M is one of the 226 coherent monomial A-graded ideals of our running example. The above set of 13 binomials in $\mathbb{C}[x, z]$ give the universal family for $Hilb_A$ around this M.

The local chart of $Hilb_A$ around the point M is a subscheme of affine space $\mathbb{C}^{13}$ with coordinates $z_1, \ldots, z_{13}$, whose defining equations are obtained as follows: Extend the weight vector w by assigning weight zero to all variables z_i, so that the first term in each of the above 13 binomials is the leading term. For each pair of binomials corresponding to a minimal syzygy of M, form their S-pair and then reduce it to a normal form with respect to the 13 binomials above. For instance,

$$S\big(c^5 - z_{10}ab^3d, ce^5 - z_{11}d^6\big) \;=\; z_{11}c^4d^6 - z_{10}ab^3de^5 \;\longrightarrow\; b^4d^2e^4\cdot(z_2^4 z_{11} - z_1 z_{10}).$$

Each such normal form is a monomial in a, b, c, d, e times a binomial in $z_1, \ldots, z_{13}$. The set of all these binomials, in the z-variables, generates the ideal J_M of local equations of $Hilb_A$ around M. In our example, J_M is generated by 27 nonzero binomials. This computation can be done in *Macaulay 2* using the procedure `localCoherentEquations`.

```
i47 : localCoherentEquations = (IA) -> (
           -- IA is the toric ideal of A living in a ring equipped
           -- with weight order w, if we are computing the local
           -- equations about the initial ideal of IA w.r.t. w.
           R := ring IA;
           w := (monoid R).Options.Weights;
           M := ideal leadTerm IA;
           S := first entries ((gens M) % IA);
           -- Make the universal family J in a new ring.
           nv := numgens R; n := numgens M;
           T = (coefficientRing R)[generators R, z_1 .. z_n,
                                Weights => flatten splice{w, n:0},
                                MonomialSize => 16];
           M = substitute(generators M,T);
           S = apply(S, s -> substitute(s,T));
           J = ideal apply(n, i ->
                     M_(0,i) - T_(nv + i) * S_i);
           -- Find the ideal Ihilb of local equations about M:
           spairs := (gens J) * (syz M);
           g := forceGB gens J;
           B = (coefficientRing R)[z_1 .. z_n,MonomialSize=>16];
           Fones := map(B,T, matrix(B,{splice {nv:1}}) | vars B);
           Ihilb := ideal Fones (spairs % g);
           Ihilb
           );
```

Suppose we wish to calculate the local equations about $M = in_w(I_A)$. The input to `localCoherentEquations` is the toric ideal I_A living in a polynomial ring equipped with the weight order specified by w. This is done as follows:

```
i48 : IA = toricIdeal A;

o48 : Ideal of R

i49 : Y = QQ[a..e, MonomialSize => 16,
                 Degrees => transpose A, Weights => {9,3,5,0,0}];

i50 : IA = substitute(IA,Y);

o50 : Ideal of Y
```

The initial ideal M is calculated in the third line of the algorithm, and `S` stores the standard monomials of M of the same degrees as the minimal generators of M. We could have calculated `S` using our old procedure `stdMonomials` but this involves computing the monomials in R_b for various values of b which can be slow on large examples. As by-products, `localCoherentEquations` also gets `J`, the ideal of the universal family for $Hilb_A$ about M, the ring `T` of this ideal, and the ring `B` of `Ihilb` which is the ideal of the affine patch of $Hilb_A$ about M. The matrix `spairs` contains all the S-pairs between generators of `J` corresponding to the minimal first syzygies of M. The command `forceGB` is used to declare the generators of `J` to be a Gröbner basis, and `Fones` is the ring map from `T` to `B` that sends each of a, b, c, d, e to one and the z variables to themselves. The columns of the matrix `(spairs % g)` are the normal forms of the polynomials in `spairs` with respect to the forced Gröbner basis `g` and the ideal `Ihilb` of local equations is generated by the image of these normal forms in the ring `B` under the map `Fones`.

```
i51 : JM = localCoherentEquations(IA)

                                                                  ...
o51 = ideal (z z  - z , z z  - z , - z z  + z , - z z  + z , - z z  +  ...
              1 2    3   1 2    3     4 7    2     5 8    2     1 5    ...

o51 : Ideal of B
```

Removing duplications among the generators,

$$J_M = \langle z_1 - z_{10}z_{11}, z_2 - z_4z_7, z_2 - z_5z_8, z_2 - z_{11}z_{12}, z_2 - z_1z_{11}z_{13},\\ z_3 - z_1z_2, z_3 - z_5z_9, z_4 - z_1z_5, z_6 - z_3z_5, z_6 - z_1z_2z_5, z_7 - z_1z_{10}, z_8 - z_1z_7,\\ z_9 - z_1z_8, z_{12} - z_1z_{13}, z_1z_2 - z_5z_9, z_1z_2 - z_1z_5z_8, z_1z_2 - z_1^2z_4z_{10}, z_1z_2 - z_1^2z_5z_7,\\ z_1z_2 - z_1z_{11}z_{12}, z_1z_2 - z_2z_{10}z_{11}, z_1^3z_4 - z_3z_{11}, z_1z_5z_8 - z_4z_8, z_2z_{10} - z_1z_{12},\\ z_3z_4 - z_1z_6, z_3z_7 - z_2z_8, z_3z_8 - z_2z_9, z_3z_{10} - z_2z_7\rangle.$$

Notice that there are many generators of J_M which have a single variable as one of its terms. Using these generators we can remove variables from other binomials. This is done in *Macaulay 2* using the subroutine `removeRedundantVariables` which is the main ingredient of the package `minPres.m2` for computing the minimal presentations of polynomial quotient rings. Both `removeRedundantVariables` and `minPres.m2` are explained in Appendix 2. The command `removeRedundantVariables` applied to an ideal in a polynomial ring (not quotient ring) creates a ring map from the ring to itself that sends the redundant variables to polynomials in the non-redundant variables and the non-redundant variables to themselves. Applying this to our ideal J_M we obtain the following simplifications.

```
i52 : load "minPres.m2";

i53 : G = removeRedundantVariables JM

                            3  2      4  3                2 4  3    2  ···
o53 = map(B,B,{z   z   , z z   z   , z z   z   , z z   z   , z , z z   z   , z   ···
                10  11    5 10  11    5 10  11    5 10  11    5   5 10  11    10 ···

o53 : RingMap B <--- B

i54 : ideal gens gb(G JM)

                3  2         2
o54 = ideal(z z   z    - z   z   z  )
             5 10  11     10  11  13

o54 : Ideal of B
```

Thus our affine patch of $Hilb_A$ has the coordinate ring

$$\mathbb{C}[z_1, z_2, \ldots, z_{13}]/J_M \;\simeq\; \frac{\mathbb{C}[z_5, z_{10}, z_{11}, z_{13}]}{\langle z_5 z_{10}^3 z_{11}^2 - z_{10} z_{11}^2 z_{13}\rangle} = \frac{\mathbb{C}[z_5, z_{10}, z_{11}, z_{13}]}{\langle (z_5 z_{10}^2 - z_{13}) z_{10} z_{11}^2\rangle}.$$

Hence, we see immediately that there are three components through the point M on $Hilb_A$. The restriction of the coherent component to the affine neighborhood of M on $Hilb_A$ is defined by the ideal quotient $(J_M : (z_1 z_2 \cdots z_{13})^\infty)$ and hence the first of the above components is an affine patch of the coherent component. Locally near M it is given by the single equation $z_5 z_{10}^2 - z_{13} = 0$ in $\mathbb{A}^4$. It is smooth and, as expected, has dimension three. The second component, $z_{10} = 0$, is also of dimension three and is smooth at M. The third component, given by $z_{11}^2 = 0$ is more interesting. It has dimension three as well, but is not reduced. Thus we have proved the following result.

Proposition 6. *The toric Hilbert scheme $Hilb_A$ of the matrix*

$$A \quad = \quad \begin{pmatrix} 1\,1\,1\,1\,1 \\ 0\,1\,2\,7\,8 \end{pmatrix}$$

is not reduced.

We can use the ring map `G` from above to simplify `J` so as to involve only the four variables z_5, z_{10}, z_{11} and z_{13}.

```
i55 : CX = QQ[a..e, z_5,z_10,z_11,z_13, Weights => splice
          {9,3,5,0,0,0,0,0}];

i56 : F = map(CX, ring J, matrix{{a,b,c,d,e}} |
                  substitute(G.matrix,CX))

                                         3  2      4  3                   ···
o56 = map(CX,T,{a, b, c, d, e, z   z   , z z   z   , z z   z   , z z   z   , z ···
                                10  11    5 10  11    5 10  11    5 10  11     ···

o56 : RingMap CX <--- T
```

Applying this map to J we get the ideal $J1$,

```
i57 : J1 = F J

                                         3  2           2   4  3       ···
o57 = ideal (c*d - b*e*z  z  , a*e - b*d*z z  z  , a*c - b z z  z  , a ···
                        10 11              5 10 11          5 10 11    ···

o57 : Ideal of CX
```

and adding the ideal $\langle z_{11}^2 \rangle$ to $J1$ we obtain the universal family for the non-reduced component of $Hilb_A$ about M.

```
i58 : substitute(ideal(z_11^2),CX) + J1

              2                                 3  2           2   4   ···
o58 = ideal (z  , c*d - b*e*z  z  , a*e - b*d*z z  z  , a*c - b z z  z ···
              11             10 11              5 10 11          5 10  ···

o58 : Ideal of CX
```

In the rest of this section, we present an interpretation of the ideal J_M in terms of the combinatorial theory of *integer programming*. See, for instance, [22, §4] or [24] for the relevant background. Our reduced Gröbner basis (2) is the *minimal test set* for the family of integer programs

$$\text{Minimize} \quad w \cdot u \quad \text{subject to } A \cdot u = b \text{ and } u \in \mathbb{N}^n, \tag{4}$$

where $A \in \mathbb{N}^{d \times n}$ and $w \in \mathbb{Z}^n$ are fixed and b ranges over $\mathbb{N}^d$. If $u' \in \mathbb{N}^n$ is any feasible solution to (4), then the corresponding optimal solution $u \in \mathbb{N}^n$ is computed as follows: the monomial x^u is the unique normal form of $x^{u'}$ modulo the Gröbner basis (2).

Suppose we had reduced $x^{u'}$ modulo the binomials (3) instead of (2). Then the output has a z-factor which depends on our choice of reduction path. To be precise, suppose the reduction path has length m and at the j-th step we had used the reduction $x^{u_{\mu_j}} \to z_{\mu_j} \cdot x^{v_{\mu_j}}$. Then we would obtain the normal form

$$z_{\mu_1} z_{\mu_2} z_{\mu_3} \cdots z_{\mu_m} \cdot x^u.$$

Reduction paths can have different lengths. If we take another path which has length m' and uses $x^{u_{\nu_j}} \to z_{\nu_j} \cdot x^{v_{\nu_j}}$ at the j-th step, then the output would be

$$z_{\nu_1} z_{\nu_2} z_{\nu_3} \cdots z_{\nu_{m'}} \cdot x^u.$$

Theorem 7. *The ideal J_M of local equations on $Hilb_A$ is generated by the binomials*

$$z_{\mu_1} z_{\mu_2} z_{\mu_3} \cdots z_{\mu_m} \quad - \quad z_{\nu_1} z_{\nu_2} z_{\nu_3} \cdots z_{\nu_{m'}}$$

each encoding a pair of distinct reduction sequences from a feasible solution of an integer program of the type (4) to the corresponding optimal solution using the minimal test set in (2).

Proof: The given ideal is contained in J_M because its generators are differences of monomials arising from the possible reduction paths of $lcm(x^{u_i}, x^{u_j})$, for $1 \leq i, j \leq r$. Conversely, any reduction sequence can be transformed into an equivalent reduction sequence using S-pair reductions. This follows from standard arguments in the proof of Buchberger's criterion [5, §2.6, Theorem 6], and it implies that the binomials $z_{\mu_1} \cdots z_{\mu_m} - z_{\nu_1} \cdots z_{\nu_{m'}}$ are $\mathbb{C}[z]$-linear combinations of the generators of J_M. $\square$

A given feasible solution of an integer program (4) usually has many different reduction paths to the optimal solution using the reduced Gröbner basis (2). For our matrix 1 and cost vector $w = (9, 3, 5, 0, 0)$, the monomial a^2bde^6 encodes the feasible solution $(2, 1, 0, 1, 6)$ of the integer program

$$\text{Minimize} \quad w \cdot u \quad \text{subject to } A \cdot u = \begin{pmatrix} 10 \\ 56 \end{pmatrix} \text{ and } u \in \mathbb{N}^5.$$

There are 19 different paths from this feasible solution to the optimal solution $(0, 3, 0, 3, 4)$ encoded by the monomial $b^3d^3e^4$. The generating function for these paths is:

$$z_1^2 + 3z_1z_2^2z_5z_7 + 2z_1z_2z_5z_7^2z_{12} + 2z_1z_2z_5z_8 + 2z_1z_2z_{12}z_{13} + z_1z_5z_9$$
$$+z_2^3z_4z_5z_7^2 + z_2^3z_4z_{13} + z_2^3z_5z_{11} + 2z_2z_3z_5z_7 + z_3z_5z_7^2z_{12} + z_3z_5z_8 + z_3z_{12}z_{13}.$$

The difference of any two monomials in this generating function is a valid local equation for the toric Hilbert scheme of (1). For instance, the binomial $z_3z_5z_7^2z_{12} - z_3z_{12}z_{13}$ lies in J_M, and, conversely, J_M is generated by binomials obtained in this manner.

The scheme structure of J_M encodes obstructions to making certain reductions when solving our family of integer programs. For instance, the variable z_3 is a zero-divisor modulo J_M. If we factor it out from the binomial $z_3z_5z_7^2z_{12} - z_3z_{12}z_{13} \in J_M$, we get $z_5z_7^2z_{12} - z_{12}z_{13}$ which does not lie in J_M. Thus there is no monomial $a^{i_1}b^{i_2}c^{i_3}d^{i_4}e^{i_5}$ for which both the paths $z_5z_7^2z_{12}$ and $z_{12}z_{13}$ are used to reach the optimum. It would be a worthwhile combinatorial project to study the path generating functions and their relation to the ideal J_M in more detail.

It is instructive to note that the binomials $z_{\mu_1}z_{\mu_2} \cdots z_{\mu_m} - z_{\nu_1}z_{\nu_2} \cdots z_{\nu_{m'}}$ in Theorem 7 do not form a vector space basis for the ideal J_M. We demonstrate this for the lexicographic Gröbner basis (with $a \succ b \succ c \succ d \succ e$) of the toric ideal defining the rational normal curve of degree 4. In this case, we can take $A = \begin{pmatrix} 1\,1\,1\,1\,1 \\ 0\,1\,2\,3\,4 \end{pmatrix}$ and the universal family in question is :

$$\{ac - z_1b^2,\, ad - z_2bc,\, ae - z_3c^2,\, bd - z_4c^2,\, be - z_5cd,\, ce - z_6d^2\}.$$

The corresponding ideal of local equations is $J_M = \langle z_3 - z_2z_5, z_2 - z_1z_4, z_5 - z_4z_6\rangle$, from which we see that M is a smooth point of $Hilb_A$. The binomial $z_1z_5 - z_1z_4z_6$ lies in J_M but there is no monomial which has the reduction path z_1z_5 or z_5z_1 to optimality. Indeed, any monomial which admits the reductions z_1z_5 or z_5z_1 must be divisible by either ace or abe. The path generating functions for these two monomials are

$$abe \quad \rightarrow \quad (z_3 \,+\, z_1z_4z_5 \,+\, z_2z_5) \cdot bc^2$$

$$ace \quad \rightarrow \quad (z_3 \,+\, z_1 z_4 z_5 \,+\, z_2 z_4 z_6) \cdot c^3.$$

Thus every reduction to optimality using z_1 and z_5 must also use z_4, and we conclude that $z_1 z_5 - z_1 z_4 z_6$ is not in the $\mathbb{C}$-span of the binomials listed in Theorem 7.

4 The coherent component of the toric Hilbert scheme

In this section we study the component of the toric Hilbert scheme $Hilb_A$ that contains the point corresponding to the toric ideal I_A. An A-graded ideal is coherent if and only if it is isomorphic to an initial ideal of I_A under the action of the torus $(\mathbb{C}^*)^n$. All coherent A-graded ideals lie on the same component of $Hilb_A$ as I_A. We will show that this component need not be normal, and we will describe how its local and global equations can be computed using *Macaulay 2*. Every term order for the toric ideal I_A can be realized by a weight vector which is an element in the lattice $N = Hom_{\mathbb{Z}}(ker_{\mathbb{Z}}(A), \mathbb{Z}) \simeq \mathbb{Z}^{n-d}$. Two weight vectors w and w' in N are considered *equivalent* if they define the same initial ideal $in_w(I_A) = in_{w'}(I_A)$. These equivalence classes are the relatively open cones of a projective fan Σ_A called the *Gröbner fan* of I_A [15], [23]. This fan lies in $\mathbb{R}^{n-d}$, the real vector space spanned by the lattice N.

Theorem 8. *The toric ideal I_A lies on a unique irreducible component of the toric Hilbert scheme $Hilb_A$, called the coherent component. The normalization of the coherent component is the projective toric variety defined by the Gröbner fan of I_A.*

Proof: The *divisor at infinity* on the toric Hilbert scheme $Hilb_A$ consists of all points at which at least one of the local coordinates (around some monomial A-graded ideal) is zero. This is a proper closed codimension one subscheme of $Hilb_A$, parametrizing all those A-graded ideals which contain at least one monomial. The complement of the divisor at infinity in $Hilb_A$ consists of precisely the orbit of I_A under the action of the torus $(\mathbb{C}^*)^n$. This is the content of [22, Lemma 10.12].

The closure of the $(\mathbb{C}^*)^n$-orbit of I_A is a reduced and irreducible component of $Hilb_A$. It is reduced because I_A is a smooth point on $Hilb_A$, as can be seen from the local equations, and it is irreducible since $(\mathbb{C}^*)^n$ is a connected group. It is a component of $Hilb_A$ because its complement lies in a divisor. We call this irreducible component the *coherent component* of $Hilb_A$.

Identifying $(\mathbb{C}^*)^n$ with $Hom_{\mathbb{Z}}(\mathbb{Z}^n, \mathbb{C}^*)$, we note that the stabilizer of I_A consists of those linear forms w which restrict to zero on the kernel of A. Therefore the coherent component is the closure in $Hilb_A$ of the orbit of the point I_A under the action of the torus $N \otimes \mathbb{C}^* = Hom_{\mathbb{Z}}(ker_{\mathbb{Z}}(A), \mathbb{C}^*)$. The $(N \otimes \mathbb{C}^*)$-fixed points on this component are precisely the coherent monomial A-graded ideals, and the same holds for the toric variety of the Gröbner fan.

Fix a maximal cone σ in the Gröbner fan Σ_A, and let $M = \langle x^{u_1}, \ldots, x^{u_r} \rangle$ be the corresponding (monomial) initial ideal of I_A. As before we write

$$\{x^{u_1} - z_1 \cdot x^{v_1}\ ,\ x^{u_2} - z_2 \cdot x^{v_2}\ ,\ \ldots\ ,\ x^{u_r} - z_r \cdot x^{v_r}\}$$

for the universal family arising from the corresponding reduced Gröbner basis of I_A. Let J_M be the ideal in $\mathbb{C}[z_1, z_2, \ldots, z_r]$ defining this family.

The restriction of the coherent component to the affine neighborhood of M on $Hilb_A$ is defined by $J_M : (z_1 z_2 \cdots z_r)^\infty$. It then follows from our combinatorial description of the ideal J_M that this ideal quotient is a binomial prime ideal. In fact, it is the ideal of algebraic relations among the Laurent monomials $x^{u_1 - v_1}, \ldots, x^{u_r - v_r}$. We conclude that the restriction of the coherent component to the affine neighborhood of M on $Hilb_A$ equals

$$\text{Spec } \mathbb{C}\big[x^{u_1 - v_1}, x^{u_2 - v_2}, \ldots, x^{u_r - v_r}\big]. \tag{5}$$

The abelian group generated by the vectors $u_1 - v_1, \ldots, u_r - v_r$ equals $\ker_{\mathbb{Z}}(A) = Hom_{\mathbb{Z}}(N, \mathbb{Z})$. This follows from [21, Lemma 12.2] because the binomials $x^{u_i} - x^{v_i}$ generate the toric ideal I_A. The cone generated by the vectors $u_1 - v_1, \ldots, u_r - v_r$ is precisely the polar dual $\sigma^\vee$ to the Gröbner cone σ. This follows from equation (2.6) in [21]. We conclude that the normalization of the affine variety (5) is the normal affine toric variety

$$\text{Spec } \mathbb{C}\big[\ker_{\mathbb{Z}}(A) \cap \sigma^\vee\big]. \tag{6}$$

The normalization morphism from (6) to (5) maps the identity point in the toric variety (6) to the point I_A in the affine chart (5) of the toric Hilbert scheme $Hilb_A$. Clearly, this normalization morphism is equivariant with respect to the action by the torus $N \otimes \mathbb{C}^*$. These two properties hold for every maximal cone σ of the Gröbner fan Σ_A. Hence there exists a unique $N \otimes \mathbb{C}^*$-equivariant morphism ϕ from the projective toric variety associated with Σ_A onto the coherent component of $Hilb_A$, such that ϕ maps the identity point to the point I_A on $Hilb_A$, and ϕ restricts to the normalization morphism (6) $\to$ (5) on each affine open chart. We conclude that ϕ is the desired normalization map from the projective toric variety associated with the Gröbner fan of I_A onto the coherent component of the toric Hilbert scheme $Hilb_A$. $\square$

We now present an example which shows that the coherent component of $Hilb_A$ need not be normal. This example is derived from the matrix that appears in Example 3.15 of [10]. This example is also mentioned in [17] without details. Let $d = 4$ and $n = 7$ and fix the matrix

$$A \quad = \quad \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 6 & 7 & 5 & 8 & 4 & 3 \\ 3 & 7 & 2 & 0 & 7 & 6 & 1 \\ 6 & 5 & 2 & 6 & 5 & 0 & 0 \end{pmatrix}. \tag{7}$$

The lattice $N = Hom_{\mathbb{Z}}(ker_{\mathbb{Z}}(A), \mathbb{Z})$ is three-dimensional. The toric ideal I_A is minimally generated by 30 binomials of total degree between 6 and 93.

```
i59 : A = {{1,1,1,1,1,1,1},{0,6,7,5,8,4,3},{3,7,2,0,7,6,1},
         {6,5,2,6,5,0,0}};

i60 : IA = toricIdeal A

             2 3       3 2   2     4 4    8 4   4 3 6    7 2 4     4 ···
o60 = ideal (a c e - b*d f , a c*d*e f  - b g , d e f  - b c g , a*b c ···

o60 : Ideal of R
```

We fix the weight vector $w = (0, 0, 276, 220, 0, 0, 215)$ in N and compute the initial ideal $M = in_w(I_A)$. This initial ideal has 44 minimal generators.

```
i61 : Y = QQ[a..g, MonomialSize => 16,
                Weights => {0,0,276,220,0,0,215},
                Degrees =>transpose A];

i62 : IA = substitute(IA,Y);

o62 : Ideal of Y

i63 : M = ideal leadTerm IA

             2 3    8 4   7 2 4     4 7 3   5 4 3 5   2 6 5 4   3 3 1 ···
o63 = ideal (a c e, b g , b c g , a*b c f , b c d f , a b c g , a b c  ···

o63 : Ideal of Y
```

Proposition 9. *The three dimensional affine variety (5), for the initial ideal M with respect to $w = (0, 0, 276, 220, 0, 0, 215)$ of the toric ideal of A in (7), is not normal.*

Proof. The universal family for the toric Hilbert scheme $Hilb_A$ at M is:

$$\{ a^2e^{15}g^{18} - z_1b^3c^6d^{10}f^{16},\ b^{13}d^{15}f^{16} - z_2a^8ce^{21}g^{14},\ c^{59}d^{57}f^{110} - z_3e^{92}g^{134},$$
$$ac^{14}d^{11}f^{23} - z_4be^{19}g^{29},\ b^7c^2g^4 - z_5d^4e^3f^6,\ \ldots,\ bc^{34}d^{32}f^{62} - z_{44}e^{53}g^{76}\}.$$

The semigroup algebra in (5) is generated by 44 Laurent monomials gotten from this family. It turns out that the first four monomials suffice to generate the semigroup. In other words, for all $j \in \{5, 6, \ldots, 44\}$ there exist $i_1, i_2, i_3, i_4 \in \mathbb{N}$ such that $z_j - z_1^{i_1}z_2^{i_2}z_3^{i_3}z_4^{i_4} \in J_M : (z_1 \cdots z_{44})^\infty$. Hence the semigroup algebra in (5) is:

$$\mathbb{C}[\frac{a^2e^{15}g^{18}}{b^3c^6d^{10}f^{16}}, \frac{b^{13}d^{15}f^{16}}{a^8ce^{21}g^{14}}, \frac{c^{59}d^{57}f^{110}}{e^{92}g^{134}}, \frac{ac^{14}d^{11}f^{23}}{be^{19}g^{29}}] \ \simeq\ \frac{\mathbb{C}[z_1, z_2, z_3, z_4]}{\langle z_1^5z_2z_3 - z_4^2\rangle}.$$

This algebra is not integrally closed, since a toric hypersurface is normal if and only if at least one of the two monomials in the defining equation is square-free. Its integral closure in $\mathbb{C}[ker_{\mathbb{Z}}(A)]$ is generated by the Laurent monomial

$$\frac{z_4}{z_1^2} \ =\ (z_1z_2z_3)^{\frac{1}{2}} \ =\ \frac{b^5c^{26}d^{31}f^{55}}{a^3e^{49}g^{65}}. \tag{8}$$

Hence the affine chart (6) of the toric variety of the Gröbner fan of I_A is the spectrum of the normal domain $\mathbb{C}[z_1, z_2, z_3, y]/\langle z_1 z_2 z_3 - y^2\rangle$, where y maps to (8). □

We now examine the local equations of $Hilb_A$ about M for this example.

```
i64 : JM = localCoherentEquations(IA)

                                                                     ...
o64 = ideal (z z  - z , z z  - z , z z  - z , z z  - z , z z  - z , z ...
              1 2    3   1 2    3   1 5    4   1 3    6   1 3    6   1 ...

o64 : Ideal of B

i65 : G = removeRedundantVariables JM;

o65 : RingMap B <--- B

i66 : toString ideal gens gb(G JM)

o66 = ideal(z_32*z_42^2*z_44-z_37^2*z_42,z_32^3*z_35*z_37^2-z_42^2*z_4 ...

o66 : String
```

This ideal has six generators and decomposing it we see that there are five components through the monomial ideal M on this toric Hilbert scheme. They are defined by the ideals:

– $\langle z_{32}z_{42}z_{44} - z_{37}^2, z_{32}^4 z_{35} - z_{42}, z_{32}^3 z_{35} z_{37}^2 - z_{42}^2 z_{44}, z_{32}^2 z_{35} z_{37}^4 - z_{42}^3 z_{44}^2,$ $z_{32} z_{35} z_{37}^6 - z_{42}^4 z_{44}^3, z_{35} z_{37}^8 - z_{42}^5 z_{44}^4\rangle$
– $\langle z_{44}, z_{37}\rangle$
– $\langle z_{37}, z_{42}^2\rangle$
– $\langle z_{42}, z_{35}\rangle$
– $\langle z_{42}, z_{32}^3\rangle$.

All five components are three dimensional. The first component is an affine patch of the coherent component and two of the components are not reduced. Let K be the first of these ideals.

```
i67 : K = ideal(z_32*z_42*z_44-z_37^2,z_32^4*z_35-z_42,
        z_32^3*z_35*z_37^2-z_42^2*z_44,z_32^2*z_35*z_37^4-z_42^3*z_4 ...
        z_32*z_35*z_37^6-z_42^4*z_44^3,z_35*z_37^8-z_42^5*z_44^4);

o67 : Ideal of B
```

Applying `removeRedundantVariables` to K we see that the affine patch of the coherent component is, locally at M, a non-normal hypersurface singularity (agreeing with (8)). The labels on the variables depend on the order of elements in the initial ideal M computed by *Macaulay 2* in line `i61`.

```
i68 : GG = removeRedundantVariables K;

o68 : RingMap B <--- B

i69 : ideal gens gb (GG K)

             5         2
o69 = ideal(z  z  z   - z  )
             32 35 44    37

o69 : Ideal of B
```

There is a general algorithm due to de Jong [6] for computing the normalization of any affine variety. In the toric case, the problem of normalization amounts to computing the minimal *Hilbert basis* of a given convex rational polyhedral cone [20]. An efficient implementation can be found in the software package `Normaliz` by Bruns and Koch [4].

Our computational study of the toric Hilbert scheme in this chapter was based on local equations rather than global equations (arising from a projective embedding of $Hilb_A$), because the latter system of equations tends to be too large for most purposes. Nonetheless, they are interesting. In the remainder of this section, we present a canonical projective embedding of the coherent component of $Hilb_A$.

Let $G_1, G_2, G_3, \ldots, G_s$ denote all the *Graver fibers* of the matrix A. In Section 1 we showed how to compute them in *Macaulay 2*. Each set G_i consists of the monomials in $\mathbb{C}[x_1, \ldots, x_n]$ which have a fixed Graver degree. Consider the set $\mathbf{G} := G_1 G_2 G_3 \cdots G_s$ which consists of all monomials which are products of monomials, one from each of the distinct Graver fibers. Let t denote the cardinality of $\mathbf{G}$. We introduce an extra indeterminate z, and we consider the $\mathbb{N}$-graded semigroup algebra $\mathbb{C}[z\mathbf{G}]$, which is a subalgebra of $\mathbb{C}[x_1, \ldots, x_n, z]$. The grading of this algebra is $deg(z) = 1$ and $deg(x_i) = 0$. Labeling the elements of $\mathbf{G}$ with indeterminates y_i, we can write

$$\mathbb{C}[z\mathbf{G}] \quad = \quad \mathbb{C}[y_1, y_2, \ldots, y_t]/P_A,$$

where P_A is a homogeneous toric ideal associated with a configuration of t vectors in $\mathbb{Z}^{n+1}$. We note that the torus $(\mathbb{C}^*)^n$ acts naturally on $\mathbb{C}[z\mathbf{G}]$.

Example 10. Let $n = 4, d = 2$ and $A = \begin{pmatrix} 3\,2\,1\,0 \\ 0\,1\,2\,3 \end{pmatrix}$, so that I_A is the ideal of the twisted cubic curve. There are five Graver fibers:

```
i70 : A = {{1,1,1,1},{0,1,2,3}};

i71 : I = toricIdeal A;

o71 : Ideal of R

i72 : Graver = graver I;

                 1       5
o72 : Matrix R  <--- R

i73 : fibers = graverFibers Graver;

i74 : peek fibers

o74 = HashTable{{2, 2} => {0, 0} | ac b2 |      }
                {2, 3} => {0, 0} | ad bc |
                {2, 4} => {0, 0} | bd c2 |
                {3, 3} => {0, 0} | a2d abc b3 |
                {3, 6} => {0, 0} | ad2 bcd c3 |

o74 : Net
```

The set $\mathbf{G} = G_1 G_2 G_3 G_4 G_5$ consists of 22 monomials of degree 14.

```
i75 : G = trim product(values fibers, ideal)

              5     5   4 3 5   5 3 4   4 2 2 4   3 4   4   2 6 4   4 ···
o75 = ideal (a b*c*d , a b d , a c d , a b c d , a b c*d , a b d , a b ···

o75 : Ideal of R

i76 : numgens G

o76 = 22
```

We introduce a polynomial ring in 22 variables $y_1, y_2, \ldots, y_{22}$, and we compute the ideal P_A. It is generated by 180 binomial quadrics.

```
i77 : z = symbol z;

i78 : S = QQ[a,b,c,d,z];

i79 : zG = z ** substitute(gens G, S);

              1       22
o79 : Matrix S  <--- S

i80 : R = QQ[y_1 .. y_22];

i81 : F = map(S,R,zG)

                5     5    4 3 5    5 3 4    4 2 2 4    3 4    4     2 6 ···
o81 = map(S,R,{a b*c*d z, a b d z, a c d z, a b c d z, a b c*d z, a b  ···

o81 : RingMap S <--- R

i82 : PA = trim ker F

              2                                                        ···
o82 = ideal (y   - y  y  , y  y   - y  y  , y  y   - y  y  , y  y   -  ···
              21    20 22   19 21    18 22   18 21    17 22   17 21    ···

o82 : Ideal of R
```

These equations define a toric surface of degree 30 in projective 21-space.

```
i83 : codim PA

o83 = 19

i84 : degree PA

o84 = 30
```

The surface is smooth, but there are too many equations and the codimension is too large to use the Jacobian criterion for smoothness [7, §16.6] directly. Instead we check smoothness for each open set $y_i \neq 0$.

```
i85 : Aff = apply(1..22, v -> (
                              K = substitute(PA,y_v => 1);
                              FF = removeRedundantVariables K;
                              ideal gens gb (FF K)));
```

```
i86 : scan(Aff, i -> print toString i);
ideal()
ideal()
ideal()
ideal(y_1^4*y_5*y_21-1)
ideal(y_1^4*y_6^6*y_21-1)
ideal()
ideal(y_1^2*y_11^2*y_17-1)
ideal(y_1^3*y_9^2*y_21^2-1)
ideal(y_6^3*y_21-y_10,y_1*y_10^3-y_6^2,y_1*y_6*y_10^2*y_21-1)
ideal(y_6*y_15-1,y_2*y_15^2-y_6*y_14,y_6^2*y_14-y_2*y_15)
ideal()
ideal(y_11*y_13-1,y_1^2*y_21^3-y_13^2)
ideal(y_1^2*y_14^3*y_21^3-1)
ideal(y_10^2*y_21-1,y_1*y_15^4-y_10^3)
ideal()
ideal(y_11*y_20-1,y_3*y_20^2-y_11*y_17,y_11^2*y_17-y_3*y_20)
ideal(y_11*y_18*y_21-1,y_1*y_21^3-y_11*y_18^2,y_11^2*y_18^3-y_1*y_21^2 ...
ideal(y_1*y_19^4*y_21^4-1)
ideal(y_15*y_22-1)
ideal()
ideal(y_20*y_22-1)
ideal()
```

By examining these local equations, we see that $Hilb_A$ is smooth, and also that there are eight fixed points under the action of the 2-dimensional torus. They correspond to the variables $y_1, y_2, y_3, y_6, y_{11}, y_{15}, y_{20}$ and y_{22}. By setting any of these eight variables to 1 in the 180 quadrics above, we obtain an affine variety isomorphic to the affine plane.

Theorem 11. *The coherent component of the toric Hilbert scheme $Hilb_A$ is isomorphic to the projective spectrum $Proj\,\mathbb{C}[z\mathbf{G}]$ of the algebra $\mathbb{C}[z\mathbf{G}]$.*

Proof: The first step is to define a morphism from $Hilb_A$ to the $(t-1)$-dimensional projective space $\mathbb{P}(\mathbf{G}) = Proj\,\mathbb{C}[y_1, y_2, \ldots, y_t]$. Consider any point I on $Hilb_A$. We intersect the ideal I with the finite-dimensional vector space $\mathbb{C}G_i$, consisting of all homogeneous polynomials in $\mathbb{C}[x_1, \ldots, x_n]$ which lie in the i-th Graver degree. The definition of A-graded ideal implies that $I \cap \mathbb{C}G_i$ is a linear subspace of codimension 1 in $\mathbb{C}G_i$. We represent this subspace by an equation $g_i(I) = \sum_{u \in G_i} c_u x^u$ which is unique up to scaling. Taking the product of these polynomials for $i = 1, \ldots, t$, we get a unique (up to scaling) polynomial which is supported on $\mathbf{G} = G_1 G_2 \cdots G_t$. The map $I \mapsto g_1(I) g_2(I) \cdots g_t(I)$ defines a morphism from $Hilb_A$ to $\mathbb{P}(\mathbf{G})$. This morphism is equivariant with respect to the $(\mathbb{C}^*)^n$-action on both schemes.

Consider the restriction of this equivariant morphism to the coherent component of the toric Hilbert scheme. It maps the $(\mathbb{C}^*)^n$-orbit of the toric ideal I_A into the subvariety $Proj\,\mathbb{C}[z\mathbf{G}]$ of $\mathbb{P}(\mathbf{G})$. This inclusion is an isomorphism onto the dense torus, as the dimension of the Newton polytope of

$$g(I_A) \quad = \quad \prod_{i=1}^{t} \Big(\sum_{u \in G_i} x^u \Big)$$

equals the dimension of the kernel of A. Equivalently, the stabilizer of $g(I_A)$ in $(\mathbb{C}^*)^n$ consists only of those one-parameter subgroups w which restrict to zero on the kernel of A.

To show that our morphism is an isomorphism between the coherent component and $Proj\, \mathbb{C}[z\mathbf{G}]$, we consider the affine chart around an initial monomial ideal $M = in_w(I_A)$. The polynomial $g(M)$ is a monomial, namely, it is the product of all standard monomials whose degree is a Graver degree. Moreover, $g(M)$ is the leading monomial of $g(I_A)$ with respect to the weight vector w. The Newton polytope of $g(I_A)$ is the Minkowski sum of the Newton polytopes of the polynomials $g_1(I_A), \ldots, g_t(I_A)$, and it is a state polytope for I_A, by [22, Theorem 7.5].

Let $g(M) = x^q$, and let σ be the cone of the Gröbner fan Σ_A which has w in its interior. Then σ coincides with the normal cone at the vertex q of the state polytope described above [22, §3]. Consider the restriction of our morphism to the affine chart around M of the coherent component, as described in (5). This restriction defines an isomorphism onto the variety

$$Spec\, \mathbb{C}[\, x^{p-q} \ : \ x^p \in \mathbf{G}\,] \tag{9}$$

On the other hand, the semigroup algebra in (9) is isomorphic to that in (5) because each pair of vectors $\{u_i, v_i\}$ seen in the reduced Gröbner basis lies in one of the Graver fibers G_j. Hence our morphism restricts to an isomorphism from the affine chart around M of the coherent component onto (9). Finally, note that (9) is the principal affine open subset of $Proj\, \mathbb{C}[z\mathbf{G}]$ defined by the coordinate x^q. Hence we get an isomorphism between the coherent component of $Hilb_A$ and $Proj\, \mathbb{C}[z\mathbf{G}]$. □

5 Appendix 1: Fourier-Motzkin Elimination

We now give the *Macaulay 2* code for converting the generator/inequality representation of a rational convex polyhedron to the other. It is based on the Fourier-Motzkin elimination procedure for eliminating a variable from a system of inequalities [25]. This code was written by Greg Smith.

Given any cone $C \subset \mathbb{R}^d$, the polar cone of C is defined to be

$$C^\vee = \{x \in \mathbb{R}^d \mid x \cdot y \leq 0, \text{for all } y \in C\}.$$

For a $d \times n$ matrix Z, define $cone(Z) = \{Zx \mid x \in \mathbb{R}^n_{\geq 0}\} \subset \mathbb{R}^d$, and $affine(Z) = \{Zx \mid x \in \mathbb{R}^n\} \subset \mathbb{R}^d$. For two integer matrices Z and H, both having d rows, `polarCone(Z,H)` returns a list of two integer matrices `{A,E}` such that

$$cone(Z) + affine(H) = \{x \in \mathbb{R}^d \mid A^t x \leq 0, E^t x = 0\}.$$

Equivalently, $(cone(Z) + affine(H))^\vee = cone(A) + affine(E)$.

We now describe each routine in the package `polarCone.m2`. We have simplified the code for readability, sometimes at the cost of efficiency. We start with three simple subroutines: `primitive`, `toZZ`, and `rotateMatrix`.

The routine `primitive` takes a list of integers `L`, and divides each element of this list by their greatest common denominator.

```
i87 : primitive = (L) -> (
          n := #L-1; g := L#n;
          while n > 0 do (n = n-1; g = gcd(g, L#n);
               if g === 1 then n = 0);
          if g === 1 then L else apply(L, i -> i // g));
```

The routine toZZ converts a list of rational numbers to a list of integers, by multiplying by their common denominator.

```
i88 : toZZ = (L) -> (
          d := apply(L, e -> denominator e);
          R := ring d#0;            l := 1_R;
          scan(d, i -> (l = (l*i // gcd(l,i))));
          apply(L, e -> (numerator(l*e))));
```

The routine rotateMatrix is a kind of transpose. Its input is a matrix, and its output is a matrix of the same shape as the transpose. It places the matrix in the form so that in the routine polarCone, computing a Gröbner basis will do the Gaussian elimination that is needed.

```
i89 : rotateMatrix := (M) -> (
          r := rank source M;        c := rank target M;
          matrix table(r, c, (i,j) -> M_(c-j-1, r-i-1)));
```

The procedure of Fourier-Motzkin elimination as presented by Ziegler in [25] is used, together with some heuristics which he presents as exercises. The following, which is a kind of S-pair criterion for inequalities, comes from Exercise 2.15(i) in [25].

The routine isRedundant determines if a row vector (inequality) is redundant. Its input argument V is the same input that is used in fourierMotzkin: it is a list of sets of integers. Each entry contains indices of the original rays which do *not* vanish at the corresponding row vector. vert is a set of integers; the original rays for the row vector in question. A boolean value is returned.

```
i90 : isRedundant = (V, vert) -> (
            -- the row vector is redundant iff 'vert' contains an
            -- entry in 'V'.
            x := 0; k := 0;
            numRow := #V; -- equals the number of inequalities
            while x < 1 and k < numRow do (
                 if isSubset(V#k, vert) then x = x+1;
                 k = k+1;);
            x === 1);
```

The main work horse of polarCone.m2 is the subroutine fourierMotzkin which eliminates the first variable in the inequalities A using the double description version of Fourier-Motzkin elimination. The set A is a list of lists of integers, each entry corresponding to a row vector in the system of inequalities. The argument V is a list of sets of integers. Each entry contains the indices of the original rays which do *not* vanish at the corresponding row vector in A. Note that this set is the *complement* of the set V_i appearing in exercise 2.15 in [25]. The argument spot is the integer index of the variable being eliminated.

The routine returns a list {projA,projV} where projA is a list of lists of integers. Each entry corresponds to a row vector in the projected system of inequalities. The list projV is a list of sets of integers. Each entry contains indices of the original rays which do *not* vanish at the corresponding row vector in `projA`.

```
i91 : fourierMotzkin := (A, V, spot) -> (
          -- initializing local variables
          numRow := #A;                  -- equal to the length of V
          numCol := #(A#0);              pos := {};
          neg := {};                     projA := {};
          projV := {};                   k := 0;
          -- divide the inequalities into three groups.
          while k < numRow do (
               if A#k#0 < 0 then neg = append(neg, k)
               else if A#k#0 > 0 then pos = append(pos, k)
               else (projA = append(projA, A#k);
                    projV = append(projV, V#k););
               k = k+1;);
          -- generate new irredundant inequalities.
          scan(pos, i -> scan(neg, j -> (vert := V#i + V#j;
                         if not isRedundant(projV, vert)
                         then (iRow := A#i;     jRow := A#j;
                              iCoeff := - jRow#0;
                              jCoeff := iRow#0;
                              a := iCoeff*iRow + jCoeff*jRow;
                              projA = append(projA, a);
                              projV = append(projV, vert););)));
          -- don't forget the implicit inequalities '-t <= 0'.
          scan(pos, i -> (vert := V#i + set{spot};
               if not isRedundant(projV, vert) then (
                    projA = append(projA, A#i);
                    projV = append(projV, vert);););
          -- remove the first column
          projA = apply(projA, e -> e_{1..(numCol-1)});
          {projA, projV});
```

As mentioned above, `polarCone` takes two matrices `Z`, `H`, both having d rows, and outputs a pair of matrices `A`, `E` such that $(cone(Z)+affine(H))^\vee = cone(A)+affine(E)$.

```
i92 : polarCone(Matrix, Matrix) := (Z, H) -> (
          R := ring source Z;
          if R =!= ring source H then error ("polarCone: " |
               "expected matrices over the same ring");
          if rank target Z =!= rank target H then error (
               "polarCone: expected matrices to have the " |
               "same number of rows");
          if (R =!= ZZ) then error ("polarCone: expected " |
               "matrices over 'ZZ'");
          -- expressing 'cone(Y)+affine(B)' as '{x : Ax <= 0}'
          Y := substitute(Z, QQ);     B := substitute(H, QQ);
          if rank source B > 0 then Y = Y | B | -B;
          n := rank source Y;         d := rank target Y;
          A := Y | -id_(QQ^d);
          -- computing the row echelon form of 'A'
          A = gens gb rotateMatrix A;
          L := rotateMatrix leadTerm A;
          A = rotateMatrix A;
          -- find pivots
          numRow = rank target A;                   -- numRow <= d
          i := 0;                        pivotCol := {};
          while i < numRow do (j := 0;
               while j < n+d and L_(i,j) =!= 1_QQ do j = j+1;
               pivotCol = append(pivotCol, j);
               i = i+1;);
          -- computing the row-reduced echelon form of 'A'
          A = ((submatrix(A, pivotCol))^(-1)) * A;
          -- converting 'A' into a list of integer row vectors
          A = entries A;
          A = apply(A, e -> primitive toZZ e);
          -- creating the vertex list 'V' for double description
```

```
-- and listing the variables 'T' which remain to be
-- eliminated
V := {};                  T := toList(0..(n-1));
scan(pivotCol, e -> (if e < n then (T = delete(e, T);
               V = append(V, set{e});)));
-- separating inequalities 'A' and equalities 'E'
eqnRow := {};             ineqnRow := {};
scan(numRow, i -> (if pivotCol#i >= n then
          eqnRow = append(eqnRow, i)
          else ineqnRow = append(ineqnRow, i);));
E := apply(eqnRow, i -> A#i);
E = apply(E, e -> e_{n..(n+d-1)});
A = apply(ineqnRow, i -> A#i);
A = apply(A, e -> e_(T | toList(n..(n+d-1))));
-- successive projections eliminate the variables 'T'.
if A =!= {} then scan(T, t -> (
          D := fourierMotzkin(A, V, t);
          A = D#0;          V = D#1;));
-- output formating
A = apply(A, e -> primitive e);
if A === {} then A = map(ZZ^d, ZZ^0, 0)
else A = transpose matrix A;
if E === {} then E = map(ZZ^d, ZZ^0, 0)
else E = transpose matrix E;
(A, E));
```

If the input matrix H has no columns, it can be omitted. A sequence of two matrices is returned, as above.

```
i93 : polarCone(Matrix) := (Z) -> (
        polarCone(Z, map(ZZ^(rank target Z), ZZ^0, 0)));
```

As a simple example, consider the permutahedron in $\mathbb{R}^3$ whose vertices are the following six points.

```
i94 : H = transpose matrix{
      {1,2,3},
      {1,3,2},
      {2,1,3},
      {2,3,1},
      {3,1,2},
      {3,2,1}};

                3       6
o94 : Matrix ZZ  <--- ZZ
```

The inequality representation of the permutahedron is obtained by calling `polarCone` on H: the facet normals of the polytope are the columns of the matrix in the first argument of the output. The second argument is trivial since our input is a polytope and hence there are is no non-trivial affine space contained in it. If we call `polarCone` on the output, we will get back H as expected.

```
i95 : P = polarCone(H)

o95 = (| 1  1  1  -1 -1 -5 |, 0)
       | -1 1  -5 1  -1 1  |
       | -1 -5 1  -1 1  1  |

o95 : Sequence

i96 : Q = polarCone(P_0)

o96 = (| 1 1 2 2 3 3 |, 0)
       | 2 3 1 3 1 2 |
       | 3 2 3 1 2 1 |

o96 : Sequence
```

6 Appendix 2: Minimal presentation of rings

Throughout this chapter, we have used on several occasions the simple, yet useful subroutine `removeRedundantVariables`. In this appendix, we present *Macaulay 2* code for this routine, which is the main ingredient for finding minimal presentations of quotients of polynomial rings. Our code for this routine is a somewhat simplified, but less efficient version of a routine in the *Macaulay 2* package, `minPres.m2`, written by Amelia Taylor.

The routine `removeRedundantVariables` takes as input an ideal `I` in a polynomial ring `A`. It returns a ring map `F` from `A` to itself, which sends redundant variables to polynomials in the non-redundant variables, and sends non-redundant variables to themselves. For example,

```
i97 : A = QQ[a..e];

i98 : I = ideal(a-b^2-1, b-c^2, c-d^2, a^2-e^2)

                 2              2         2       2    2
o98 = ideal (- b  + a - 1, - c  + b, - d  + c, a  - e )

o98 : Ideal of A

i99 : F = removeRedundantVariables I

                 8       4   2
o99 = map(A,A,{d  + 1, d , d , d, e})

o99 : RingMap A <--- A
```

The non-redundant variables are d and e. The image of I under F gives the elements in this smaller set of variables. We take the ideal of a Gröbner basis of the image:

```
i100 : I1 = ideal gens gb(F I)

               16     8    2
o100 = ideal(d   + 2d  - e  + 1)

o100 : Ideal of A
```

The original ideal can be written in a cleaner way as

```
i101 : ideal compress (F.matrix - vars A) + I1

               8           4       2       16      8    2
o101 = ideal (d  - a + 1, d  - b, d  - c, d   + 2d  - e  + 1)

o101 : Ideal of A
```

Let us now describe the *Macaulay 2* code. The subroutine `findRedundant` takes a polynomial f, and finds a variable x_i in the ring of f, such that $f = cx_i + g$, for a non-zero constant c, and a polynomial g which does not involve the variable x_i. If there is no such variable, `null` is returned. Otherwise, if x_i is the first such variable , the list $\{i, c^{-1}g\}$ is returned.

```
i102 : findRedundant=(f)->(
            A := ring(f);
            p := first entries contract(vars A,f);
            i := position(p, g -> g != 0 and first degree g === 0);
            if i === null then
                null
            else (
                 v := A_i;
```

```
          c := f_v;
          {i,(-1)*(c^(-1)*(f-c*v))}
          )
     );
```

The main function `removeRedundantVariables` requires an ideal in a polynomial ring (not a quotient ring) as input. The internal routine `findnext` finds the first entry of the (one row) matrix `M` which contains a redundancy. This redundancy is used to modify the list `xmap`, which contains the images of the redundant variables. The matrix `M`, and the list `xmap` are both updated, and then we continue to look for more redundancies.

```
i103 : removeRedundantVariables = (I) -> (
            A := ring I;
            xmap := new MutableList from gens A;
            M := gens I;
            findnext := () -> (
                 p := null;
                 next := 0;
                 done := false;
                 ngens := numgens source M;
                 while next < ngens and not done do (
                   p = findRedundant(M_(0,next));
                   if p =!= null then
                        done = true
                   else next=next+1;
                 );
                 p);
            p := findnext();
            while p =!= null do (
                 xmap#(p#0) = p#1;
                 F1 := map(A,A,toList xmap);
                 F2 := map(A,A, F1 (F1.matrix));
                 xmap = new MutableList from first entries F2.matrix;
                 M = compress(F2 M);
                 p = findnext();
                 );
            map(A,A,toList xmap));
```

References

1. V.I. Arnold. A-graded algebras and continued fractions. *Communications in Pure and Applied Mathematics*, 42:993–1000, 1989.
2. A Bigatti, R. La Scala, and L. Robbiano. Computing toric ideals. *Journal of Symbolic Computation*, 27:351–365, 1999.
3. L. J. Billera, P. Filliman, and B. Sturmfels. Constructions and complexity of secondary polytopes. *Advances in Mathematics*, 83:155–179, 1990.
4. W. Bruns and R. Koch. Normaliz, a program to compute normalizations of semigroups. available by anonymous ftp from ftp.mathematik.Uni-Osnabrueck.DE/pub/osm/kommalg/software/.
5. D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra.* Springer-Verlag, New York, 1997.
6. T. de Jong. An algorithm for computing the integral closure. *Journal of Symbolic Computation*, 26:273–277, 1998.
7. D. Eisenbud. *Commutative Algebra with a View Toward Algebraic Geometry.* Springer-Verlag, New York, 1994.
8. I. M. Gel'fand, M. Kapranov, and A. Zelevinsky. *Multidimensional Determinants, Discriminants and Resultants.* Birkhäuser, Boston, 1994.

9. J.E. Graver. On the foundations of linear and integer programming. *Mathematical Programming*, 8:207–226, 1975.
10. S. Hoşten and D. Maclagan. The vertex ideal of a lattice. Preprint 2000.
11. S. Hoşten and J. Shapiro. Primary decomposition of lattice basis ideals. *Journal of Symbolic Computation*, 29:625–639, 2000.
12. B. Huber and R.R. Thomas. Computing Gröbner fans of toric ideals. *Experimental Mathematics*, 9:321–331, 2000. Software : `TiGERS` available at `http://www.math.washington.edu/~thomas/programs.html`.
13. E. Korkina, G. Post, and M. Roelofs. Classification of generalized A-graded algebras with 3 generators. *Bulletin de Sciences Mathématiques*, 119:267–287, 1995.
14. D. Maclagan and R.R. Thomas. Combinatorics of the toric Hilbert scheme. *Discrete and Computational Geometry*. To appear.
15. T. Mora and L. Robbiano. The Gröbner fan of an ideal. *Journal of Symbolic Computation*, 6:183–208, 1998.
16. I. Peeva and M. Stillman. Local equations for the toric Hilbert scheme. *Advances in Applied Mathematics*. To appear.
17. I. Peeva and M. Stillman. Toric Hilbert schemes. Preprint 1999.
18. V. Reiner. The generalized Baues problem. In L. Billera, A. Björner, C. Greene, R. Simion, and R. Stanley, editors, *New Perspectives in Algebraic Combinatorics*. Cambridge University Press, 1999.
19. F. Santos. A point configuration whose space of triangulations is disconnected. *Journal of the American Math. Soc.*, 13:611–637, 2000.
20. A. Schrijver. *Theory of Linear and Integer Programming*. Wiley-Interscience, Chichester, 1986.
21. B. Sturmfels. The geometry of A-graded algebras. math.AG/9410032.
22. B. Sturmfels. *Gröbner Bases and Convex Polytopes*, volume 8. American Mathematical Society, University Lectures, 1996.
23. B. Sturmfels and R.R. Thomas. Variation of cost functions in integer programming. *Mathematical Programming*, 77:357–387, 1997.
24. R.R. Thomas. Applications to integer programming. In D.A. Cox and B. Sturmfels, editors, *Applications of Computational Algebraic Geometry*. AMS Proceedings of Symposia in Applied Mathematics, 1997.
25. G. Ziegler. *Lectures on Polytopes*, volume 152. Springer-Verlag, New York, 1995.