Toric

toric varieties and some combinatorial geometry computations

1.9.5

7 October 2019

David Joyner

David Joyner
Email: wdjoyner@gmail.com
Homepage: https://sites.google.com/site/wdjoyner/
Address: W. David Joyner
Mathematics Department
U. S. Naval Academy
Annapolis, MD 21402
USA

Copyright

Acknowledgements

The code for the toric package was written during the summer of 2002. It was put into GAP package format in the summer of 2004.

toric is free software; you can redistribute it and/or modify it under the terms of the MIT License.

toric is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the MIT License for more details.

This documentation was prepared with the GAPDoc package of Frank Lübeck and Max Neunhöffer. Moreover, a bug in toric 1.8 was fixed with the help of Max Horn, and this documentation was modified accordingly. Finally, I thank Alexander Konovalov and Max Horn for transferring this package to the new Git repository.

Contents

Chapter 1

Introduction

1.1 Introduction to the toric package

This manual describes the toric package for working with toric varieties in GAP. Toric varieties can be dealt with more easily than general varieties since often times questions about a toric variety can be reformulated in terms of combinatorial geometry. Some coding theory commands related to toric varieties are contained in the error-correcting codes GUAVA package (for example, the command `ToricCode`). We refer to the GUAVA manual [JFM] and the expository paper [JV02] for more details.

The toric package also contains several commands unrelated to toric varieties (mostly for list manipulations). These will not be described in this documention but they are briefly documented in the `lib/util.gd` file.

toric is implemented in the GAP language, and runs on any system supporting GAP4.3 and above. The toric package is loaded with the command

```
gap> LoadPackage( "toric" );
```

Please send bug reports, suggestions and other comments about toric to support@gap-system.org.

1.2 Introduction to constructing toric varieties

Rather than sketch the theory of toric varieties, we refer to [JV02] and [Ful93] for details. However, we briefly describe some terminology and notation.

1.2.1 Generalities

Let F denote a field and $R = F[x_1,...,x_n]$ be a ring in n variables. A BINOMIAL EQUATION in R is one of the form

$$x_1^{k_1}...x_n^{k_n} = x_1^{\ell_1}...x_n^{\ell_n},$$

where $k_i \geq 0$, $\ell_j \geq 0$ are integers. A binomial variety is a subvariety of affine n-space A_F^n defined by a finite set of binomial equations (such a variety need not be normal). A typical "toric variety" is binomial, though they will be introduced via an *a priori* independent construction. The basic idea of the construction is to replace each such binomial equation as above by a relation in a semigroup contained in a lattice and replace R by the "group algebra" of this semigroup. By the way, a toric variety is always normal (see for example, [Ful93], page 29).

1.2.2 Basic combinatorial geometry constructions

Let Q denote the field of rational numbers and Z denote the set of integers. Let $n > 1$ denote an integer.

Let $V = Q^n$ having basis $f_1 = (1,0,...,0)$, ..., $f_n = (0,...,0,1)$. Let $L_0 = Z^n \subset V$ be the standard lattice in V. We identify V and $L_0 \otimes_Z Q$. We use $\langle\ ,\ \rangle$ to denote the (standard) inner product on V. Let

$$L_0^* = \mathrm{Hom}(L_0, Z) = \{v \in V \mid \langle v, w\rangle \in Z,\ \forall w \in L_0\}$$

denote the DUAL LATTICE, so (fixing the standard basis e_1^*,...,e_n^* dual to the f_1,...,f_n) L_0^* may be identified with Z^n.

A CONE in V is a set σ of the form

$$\sigma = \{a_1 v_1 + ... + a_m v_m \mid a_i \geq 0\} \subset V,$$

where $v_1, ..., v_m \in V$ is a given collection of vectors, called (semigroup) GENERATORS of σ. A RATIONAL CONE is one where $v_1, ..., v_m \in L_0$. A STRONGLY CONVEX cone is one which contains no lines through the origin.

By abuse of terminology, from now on a CONE *of L_0 is a strongly convex rational cone.*

A FACE of a cone σ is either σ itself or a subset of the form $H \cap \sigma$, where H is a codimension one subspace of V which intersects the cone non-trivially and such that the cone is contained in exactly one of the two half-spaces determined by H. A RAY (or edge) of a cone is a one-dimensional face. Typically, cones are represented in toric by the list of vectors defining their rays. The DIMENSION of a cone is the dimension of the vector space it spans. The toric package can test if a given vector is in a given cone (see `InsideCone`).

If σ is a cone then the DUAL CONE is defined by

$$\sigma^* = \{w \in L_0^* \otimes Q \mid \langle v, w\rangle \geq 0,\ \forall v \in \sigma\}.$$

The toric package can test if a vector is in the dual of a given cone (see `InDualCone`).

Associate to the dual cone σ^* is the semigroup

$$S_\sigma = \sigma^* \cap L_0^* = \{w \in L_0^* \mid \langle v, w\rangle \geq 0,\ \forall v \in \sigma\}.$$

Though L_0^* has \$n\$ generators *as a lattice*, typically S_σ will have more than n generators *as a semigroup*. The toric package can compute a minimal list of semigroup generators of S_σ (see `DualSemigroupGenerators`).

A fan is a collection of cones which "fit together" well. A FAN in L_0 is a set $\Delta = \{\sigma\}$ of rational strongly convex cones in $V = L_0 \otimes Q$ such that

- if $\sigma \in \Delta$ and $\tau \subset \sigma$ is a face of σ then $\tau \in \Delta$,
- if $\sigma_1, \sigma_2 \in \Delta$ then the intersection $\sigma_1 \cap \sigma_2$ is a face of both σ_1 and σ_2 (and hence belongs to Δ).

In particular, the face of a cone in a fan is a cone is the fan.

If V is the (set-theoretic) union of the cones in Δ then we call the fan COMPLETE. We shall assume that all fans are finite. A fan is determined by its list of maximal cones.

Notation: A fan Δ is represented in toric as a set of maximal cones. For example, if Δ is the fan with maximal cones $\sigma_1 = Q_+ \cdot f_1 + Q_+ \cdot (-f_1 + f_2)$, $\sigma_2 = Q_+ \cdot (-f_1 + f_2) + Q_+ \cdot (-f_1 - f_2)$, $\sigma_3 = Q_+ \cdot (-f_1 - f_2) + Q_+ \cdot f_1$, then Δ is represented by $[[[1,0],[-1,1]],[[-1,1],[-1,-1]],[[-1,-1],[1,0]]]$.

The toric package can compute all cones in a fan of a given dimension (see `ConesOfFan`). Moreover, toric can compute the set of all normal vectors to the faces (i.e., hyperplanes) of a cone (see `Faces`).

The STAR of a cone σ in a fan Δ is the set Δ_σ of cones in Δ containing σ as a face. The toric package can compute stars (see `ToricStar`).

1.2.3 Basic affine toric variety constructions

Let

$$R_\sigma = F[S_\sigma]$$

denote the "group algebra" of this semigroup. It is a finitely generated commutative F-algebra. It is in fact integrally closed ([Ful93], page 29). We may interprete R_σ as a subring of $R = F[x_1,...,x_n]$ as follows: First, identify each e_i^* with the variable x_i. If S_σ is generated as a semigroup by vectors of the form $\ell_1 e_1^* + ... + \ell_n e_n^*$, where ℓ_i is an integer, then its image in R is generated by monomials of the form $x_1^{\ell_1} \ldots x_n^{\ell_n}$. The toric package can compute these generating monomials (see `EmbeddingAffineToricVariety`). See Lemma 2.14 in [JV02] for more details. This embedding can also be used to resolve singularities - see section 5 of [JV02] for more details.

Let

$$U_\sigma = \text{Spec } R_\sigma.$$

This defines an AFFINE TORIC VARIETY (associated to σ). It is known that the coordinate ring R_σ of the affine toric variety U_σ has the form $R_\sigma = F[x_1,...,x_n]/J$, where J is an ideal. The toric package can compute generators of this ideal by using the `DualSemigroupGenerators` and the `EmbeddingAffineToricVariety` commands.

Roughly speaking, the toric variety $X(\Delta)$ associated to the fan Δ is given by a collection of affine pieces $U_{\sigma_1},U_{\sigma_2},\dots,U_{\sigma_d}$ which "glue" together (where $\Delta = \{\sigma_i\}$). The affine pieces are given by the zero sets of polynomial equations in some affine spaces and the gluings are given by maps $\phi_{i,j} : U_{\sigma_i} \to U_{\sigma_j}$ which are defined by ratios of polynomials on open subsets of the U_{σ_i}. The toric package does *not* compute these gluings or work directly with these (non-affine) varieties $X(\Delta)$.

A cone $\sigma \subset V$ is said to be NONSINGULAR if it is generated by part of a basis for the lattice L_0. A fan Δ of cones is said to be NONSINGULAR if all its cones are nonsingular. It is known that U_σ is nonsingular if and only if σ is nonsingular (Proposition 2.1 in [Ful93]).

EXAMPLE: In three dimensions, consider the cones $\sigma_{\varepsilon_1,\varepsilon_2,\varepsilon_3,i,j}$ generated by $(\varepsilon_1 \cdot 1, \varepsilon_2 \cdot 1, \varepsilon_3 \cdot 1)$ and the standard basis vectors f_i, f_j, where $\varepsilon_i = \pm 1$ and $1 \le i \ne j \le 3$. There are 8 cones per octant, for a total of 64 cones. Let Δ denote the fan in $V = Q^3$ determined by these maximal cones. The toric variety $X(\Delta)$ is nonsingular.

1.2.4 Riemann-Roch spaces and related constructions

Although the toric package does not work directly with the toric varieties $X(\Delta)$, it can compute objects associated with it. For example, it can compute the Euler characteristic (see `EulerCharacteristic`), Betti numbers (see `BettiNumberToric`), and the number of GF(q)-rational points (see `CardinalityOfToricVariety`) of $X(\Delta)$, *provided* Δ *is nonsingular.*

For an algebraic variety X the group of WEIL DIVISORS on X is the abelian group $Div(X)$ generated (additively) by the irreducible subvarieties of X of codimension 1. For a toric variety $X(\Delta)$ with dense open torus T, a Weil divisor D is T-INVARIANT if $D = T \cdot D$. The group of T-invariant Weil divisors is denoted $TDiv(X)$. This is finitely generated by an explicitly given finite set of divisors $\{D_1,...,D_r\}$ which correspond naturally to certain cones in Δ (see [Ful93] for details). In particular, we may represent such a divisor D in $TDiv(X)$ by an k-tuple $(d_1,...,d_k)$ of integers.

Let Δ denote a fan in $V = Q^n$ with rays (or edges) τ_i, $1 \le i \le k$, and let v_i denote the first lattice point on τ_i. Associated to the T-invariant Weil divisor $D = d_1 D_1 + ... + d_k D_k$, is the POLYTOPE

$$P_D = \{x = (x_1,...,x_n) \mid \langle x, v_i \rangle \ge -d_i,\ \forall 1 \le i \le k\}.$$

The toric package can compute P_D (see `DivisorPolytope`), as well as the set of all lattice points contained in this polytope (see `DivisorPolytopeLatticePoints`). Also associated to the T-invariant Weil divisor $D = d_1 D_1 + ... + d_k D_k$, is the Riemann-Roch space, $L(D)$. This is a space of functions on $X(\Delta)$ whose zeros and poles are "controlled" by D (for a more precise definition, see [Ful93]). The toric package can compute a basis for $L(D)$ (see `RiemannRochBasis`), *provided* Δ *is complete and nonsingular.*

Chapter 2

Cones and semigroups

2.1 Cones

This section introduces the toric commands which deal with cones and related combinatorial-geometric objects. Recall, a CONE is a strongly convex polyhedral cone ([Ful93], page 4).

2.1.1 InsideCone

▷ `InsideCone(v, L)` (function)

This command returns 'true' if the vector v belongs to the interior of the (strongly convex polyhedral) cone generated by the vectors in L.

This procedure does not check if L generates a strongly convex polyhedral cone.

Example

```
gap> L:=[[1,0,0],[1,1,0],[1,1,1],[1,0,1]];; v:=[0,0,1];;
gap> InsideCone(v,L);
false
gap> L:=[[1,0],[3,4]];;
gap> v:=[1,-7]; InsideCone(v,L);
[ 1, -7 ]
false
gap> v:=[4,-3]; InsideCone(v,L);
[ 4, -3 ]
false
gap> v:=[4,-4]; InsideCone(v,L);
[ 4, -4 ]
false
gap> v:=[4,1]; InsideCone(v,L);
[ 4, 1 ]
true
```

2.1.2 InDualCone

▷ `InDualCone(v, L)` (function)

This command returns 'true' if v belongs to the dual of the cone generated by the vectors in L.

Example

```
gap> L:=[[1,0,0],[1,1,0],[1,1,1],[1,0,1]];; v:=[0,0,1];;
gap> InDualCone(v,L);
true
gap> L:=[[1,0],[3,4]];
[ [ 1, 0 ], [ 3, 4 ] ]
gap> v:=[1,-7]; InDualCone(v,L);
[ 1, -7 ]
false
gap> v:=[4,-3]; InDualCone(v,L);
[ 4, -3 ]
true
gap> v:=[4,-4]; InDualCone(v,L);
[ 4, -4 ]
false
gap> v:=[4,1]; InDualCone(v,L);
[ 4, 1 ]
true
```

2.1.3 PolytopeLatticePoints

▷ `PolytopeLatticePoints(A, Perps)` (function)

Input: `Perps` $= [v_1,...,v_k]$ is the list of "inward normal" vectors perpendicular to the walls of a polytope P in the vector space $L_0^* \otimes Q$,
`A` $= [a_1,...,a_k]$ is a k-tuple of integers, where a_i denotes the amount the i-th "wall" (defined by the normal v_i) is shifted from the origin (each a_i is assumed non-negative).
For example, the polytope P with faces `[x=0, x=a, y=0, y=b]` has `Perps` $=$ $[[1,0],[-1,0],[0,1],[0,-1]]$ and `A` $= [0,a,0,b]$.
Output: the list of points in $P \cap L_0^*$.

Example

```
gap> Perps:=[[1,0],[-1,0],[0,1],[0,-1]];
[ [ 1, 0 ], [ -1, 0 ], [ 0, 1 ], [ 0, -1 ] ]
gap> A:=[0,4,0,3];
[ 0, 4, 0, 3 ]
gap> PolytopeLatticePoints(A,Perps);
[ [ 0, 0 ], [ 0, 1 ], [ 0, 2 ], [ 0, 3 ], [ 1, 0 ], [ 1, 1 ], [ 1, 2 ],
  [ 1, 3 ], [ 2, 0 ], [ 2, 1 ], [ 2, 2 ], [ 2, 3 ], [ 3, 0 ], [ 3, 1 ],
  [ 3, 2 ], [ 3, 3 ], [ 4, 0 ], [ 4, 1 ], [ 4, 2 ], [ 4, 3 ] ]
gap> Length(last);
20
```

2.1.4 Faces

▷ `Faces(Rays)` (function)

Input: `Rays` is a list of rays for the fan Δ
Output: All the normals to the faces (hyperplanes of the cone).

Example

```
gap> Cones1:=[[[2,-1],[-1,2]],[[-1,2],[-1,-1]],[[-1,-1],[2,-1]]];;
gap> Faces(Cones1[1]);
[ [ 1/2, 1 ], [ 2, 1 ] ]
gap> Faces(Cones1[2]);
[ [ -2, -1 ], [ -1, 1 ] ]
gap> Cones2:=[[[ 2,0,0],[0,2,0],[0,0,2]], [[2,0,0], [0,2,0], [2,-2,1],[1,2,-2]]];;
gap> Faces(Cones2[1]);
[ [ 0, 0, 1 ], [ 0, 1, 0 ], [ 1, 0, 0 ] ]
gap> Faces(Cones2[2]);
[ [ 1/3, 5/6, 1 ], [ 1/2, 0, -1 ], [ 2, 0, 1 ] ]
```

2.1.5 ConesOfFan

▷ `ConesOfFan(`*`Delta, k`*`)` (function)

Input: `Delta` is the fan of cones,
k is the dimension of the cones desired.
Output: The k-dimensional cones in the fan.

2.1.6 NumberOfConesOfFan

▷ `NumberOfConesOfFan(`*`Delta, k`*`)` (function)

Input: `Delta` is the fan of cones in $V = Q^n$,
k is the dimension of the cones counted.
Output: The number of k-dimensional cones in the fan.

Idea: The fan `Delta` is represented as a set of maximal cones. For each maximal cone, look at the k-dimensional faces obtained by taking n choose k subsets of the rays describing the cone. Certain of these k-subsets yield the desired cones.

Example

```
gap> Delta0:=[ [ [2,0,0],[0,2,0],[0,0,2] ], [ [2,0,0],[0,2,0],[2,-2,1],[1,2,-2] ] ];;
gap> NumberOfConesOfFan(Delta0,2);
6
gap> ConesOfFan(Delta0,2);
[ [ [ 0, 0, 2 ], [ 0, 2, 0 ] ], [ [ 0, 0, 2 ], [ 2, 0, 0 ] ],
  [ [ 0, 2, 0 ], [ 1, 2, -2 ] ], [ [ 0, 2, 0 ], [ 2, -2, 1 ] ],
  [ [ 0, 2, 0 ], [ 2, 0, 0 ] ], [ [ 1, 2, -2 ], [ 2, -2, 1 ] ] ]
gap> ConesOfFan(Delta0,1);
[ [ [ 0, 0, 2 ] ], [ [ 0, 2, 0 ] ], [ [ 1, 2, -2 ] ],
  [ [ 2, -2, 1 ] ], [ [ 2, 0, 0 ] ] ]
gap> NumberOfConesOfFan(Delta0,1);
5
```

2.1.7 ToricStar

▷ `ToricStar(`*`sigma, Delta`*`)` (function)

Input: `sigma` is a cone in the fan, represented by its set of maximal (i.e., highest dimensional) cones.
`Delta` is the fan of cones in $V = Q^n$.
Output: The star of the cone `sigma` in `Delta`, i.e., the cones τ which have `sigma` as a face.

Example

```
gap> MaxCones:=[ [ [2,0,0],[0,2,0],[0,0,2] ],
>                [ [2,0,0],[0,2,0],[2,-2,1],[1,2,-2] ] ];;
gap> #this is the set of maximal cones in the fan Delta
gap> ToricStar([[1,0]],MaxCones);
[  ]
gap> ToricStar([[2,0,0],[0,2,0]],MaxCones);
[ [ [ 0, 2, 0 ], [ 2, 0, 0 ] ], [ [ 2, 0, 0 ], [ 0, 2, 0 ], [ 0, 0, 2 ] ],
  [ [ 2, 0, 0 ], [ 0, 2, 0 ], [ 2, -2, 1 ], [ 1, 2, -2 ] ] ]
gap> MaxCones:=[ [ [2,0,0],[0,2,0],[0,0,2] ], [ [2,0,0],[0,2,0],[1,1,-2] ] ];;
gap> ToricStar([[2,0,0],[0,2,0]],MaxCones);
[ [ [ 0, 2, 0 ], [ 2, 0, 0 ] ], [ [ 2, 0, 0 ], [ 0, 2, 0 ], [ 0, 0, 2 ] ],
  [ [ 2, 0, 0 ], [ 0, 2, 0 ], [ 1, 1, -2 ] ] ]
gap> ToricStar([[1,0]],MaxCones);
[  ]
```

2.2 Semigroups

2.2.1 DualSemigroupGenerators

▷ `DualSemigroupGenerators(L)` (function)

Input: `L` is a list of integral n-vectors generating a cone σ.
Output: the generators of S_σ,

Idea: let M be the maximum of the absolute values of the coordinates of the `L`[i]'s, for each vector v in $[1..M]^n$, test if v is in the dual cone σ^*. If so, add v to list of possible generators. Once this for loop is finished, one can check this list for redundant generators. The trick is to simply omit those elements which are of the form $d_1 + d_2$, where d_1 and d_2 are "small" elements in the integral dual cone.

This program is not very efficient and should not be used in "large examples" involving semigroups with "many" generators. For example, if you take `L:=[[1,2,3,4],[0,1,0,7],[3,1,0,2],[0,0,1,0]];` then `DualSemigroupGenerators(L);` can exhaust GAP's memory allocation.

Example

```
gap> L:=[[1,0],[3,4]];; DualSemigroupGenerators([[1,0],[3,4]]);
[ [ 0, 0 ], [ 0, 1 ], [ 1, 0 ], [ 2, -1 ], [ 3, -2 ], [ 4, -3 ] ]
gap> L:=[[1,0,0],[1,1,0],[1,1,1],[1,0,1]];;
gap> DualSemigroupGenerators(L);
[ [ 0, 0, 0 ], [ 0, 0, 1 ], [ 0, 1, 0 ], [ 1, -1, 0 ], [ 1, 0, -1 ] ]
```

Chapter 3

Affine toric varieties

This chapter concerns toric commands which deal with the coordinate rings of affine toric varieties U_σ.

3.1 Ideals defining affine toric varieties

3.1.1 EmbeddingAffineToricVariety

▷ `EmbeddingAffineToricVariety(L)` (function)

Input: `L` is a list generating a cone (as in `DualSemigroupGenerators`).
Output: the toroidal embedding of $X = Spec(I)$, where I is the ideal of the affine toric variety (given as a list of multinomials).

Example

```
gap> phi:=EmbeddingAffineToricVariety([[1,0],[3,4]]);
[ x_2, x_1, x_1^2/x_4, x_1^3/x_4^2, x_1^4/x_4^3 ]
gap> L:=[[1,0,0],[1,1,0],[1,1,1],[1,0,1]];;
gap> phi:=EmbeddingAffineToricVariety(L);
[ x_3, x_2, x_1/x_5, x_1/x_6 ]
```

Chapter 4

Toric varieties $X(\Delta)$

This chapter concerns toric commands which deal with certain objects associated to the (non-affine) toric varieties $X(\Delta)$.

4.1 Riemann-Roch spaces

Let Δ denote a complete nonsingular fan.

4.1.1 DivisorPolytope

▷ `DivisorPolytope(D, Rays)` (function)

Input: `Rays` is the list of smallest integer vectors in the rays for the fan Δ which determine the Weil divisors of $X(\Delta)$.
`D` is the list of coefficients for the a Weil divisor.
Output: the linear expressions in the affine coordinates of the space of the cone which must be positive for a point to be in the desired polytope.

Example

```
gap> DivisorPolytope([6,6,0],[[2,-1],[-1,2],[-1,-1]]);
[ 2*x_1-x_2+6, -x_1+2*x_2+6, -x_1-x_2 ]
```

See also Example 6.13 in [JV02].

4.1.2 DivisorPolytopeLatticePoints

▷ `DivisorPolytopeLatticePoints(D, Delta, Rays)` (function)

Input: `Delta` is the fan
`Rays` is the *ordered* list of rays for `Delta`
`D` is the list of coefficients for a Weil divisor.
Output: the list of points in $P_D \cap L_0^*$ which parameterize the elements in the Riemann-Roch space $L(D)$, where P_D is the polytope associated to the divisor D (see `DivisorPolytope`).

Example

```
gap> Div:=[6,6,0];; Rays:=[[2,-1],[-1,2],[-1,-1]];;
gap> Delta0:=[[[2,-1],[-1,2]],[[-1,2],[-1,-1]],[[-1,-1],[2,-1]]];;
```

```
gap> P_Div:=DivisorPolytopeLatticePoints(Div,Delta0,Rays);
[ [ -6, -6 ], [ -5, -5 ], [ -5, -4 ], [ -4, -5 ], [ -4, -4 ], [ -4, -3 ],
  [ -4, -2 ], [ -3, -4 ], [ -3, -3 ], [ -3, -2 ], [ -3, -1 ], [ -3, 0 ],
  [ -2, -4 ], [ -2, -3 ], [ -2, -2 ], [ -2, -1 ], [ -2, 0 ], [ -2, 1 ],
  [ -2, 2 ], [ -1, -3 ], [ -1, -2 ], [ -1, -1 ], [ -1, 0 ], [ -1, 1 ],
  [ 0, -3 ], [ 0, -2 ], [ 0, -1 ], [ 0, 0 ], [ 1, -2 ], [ 1, -1 ], [ 2, -2 ] ]
```

4.1.3 RiemannRochBasis

▷ `RiemannRochBasis(D, Delta, Rays)` (function)

Input: `Delta` is a complete and nonsingular fan
`D` is the list of coefficients for the Weil divisor
`Rays` is a list of rays for the fan used to describe the Weil divisors.
Output: A basis (a list of monomials) for the Riemann-Roch space of the divisor represented by `D`.

For details on how the Weil divisors can be expressed in terms of the rays of the fan, please see section 3.3 in [Ful93]. This procedure does not check if the fan is complete and nonsingular.

Example

```
gap> Div:=[6,6,0];; Rays:=[[2,-1],[-1,2],[-1,-1]];;
gap> Delta0:=[[[2,-1],[-1,2]],[[-1,2],[-1,-1]],[[-1,-1],[2,-1]]];;
gap> RiemannRochBasis(Div,Delta0,Rays);
[ 1/(x_1^6*x_2^6), 1/(x_1^5*x_2^5), 1/(x_1^5*x_2^4), 1/(x_1^4*x_2^5),
  1/(x_1^4*x_2^4), 1/(x_1^4*x_2^3), 1/(x_1^4*x_2^2), 1/(x_1^3*x_2^4),
  1/(x_1^3*x_2^3), 1/(x_1^3*x_2^2), 1/(x_1^3*x_2), 1/x_1^3, 1/(x_1^2*x_2^4),
  1/(x_1^2*x_2^3), 1/(x_1^2*x_2^2), 1/(x_1^2*x_2), 1/x_1^2, x_2/x_1^2,
  x_2^2/x_1^2, 1/(x_1*x_2^3), 1/(x_1*x_2^2), 1/(x_1*x_2), 1/x_1, x_2/x_1,
  1/x_2^3, 1/x_2^2, 1/x_2, 1, x_1/x_2^2, x_1/x_2, x_1^2/x_2^2 ]
```

4.2 Topological invariants

Throughout this section, $X(\Delta)$ *must be non-singular*.

4.2.1 EulerCharacteristic

▷ `EulerCharacteristic(Delta)` (function)

Input: `Delta` is a nonsingular fan of cones, represented by its list of maximal cones.
Output: the Euler characteristic of the toric variety $X(\Delta)$, where Δ is a fan determined by `Delta`.

Example

```
gap> Cones:=[[[2,-1],[-1,2]],[[-1,2],[-1,-1]],[[-1,-1],[2,-1]]];;
gap> EulerCharacteristic(Cones);
3
```

Note: $X(\Delta)$ *must be non-singular* here.

4.2.2 BettiNumberToric

▷ `BettiNumberToric(Delta, k)` (function)

Input: `Delta` represents a nonsingular fan Δ (represented by maximal cones), k is an integer.
Output: the k-th Betti number of the toric variety $X(\Delta)$.

The `BettiNumberToric` procedure does not check if `Delta` is nonsingular. It is possible that this procedure outputs nonsense when `Delta` is not represented by maximal cones or is nonsingular.

Example

```
gap> Cones:=[[[2,-1],[-1,2]],[[-1,2],[-1,-1]],[[-1,-1],[2,-1]]];;
gap> BettiNumberToric(Cones,1);
0
gap> BettiNumberToric(Cones,2);
1
gap> Cones:=[[[2,-1],[-1,1]],[[-1,1],[-1,0]],[[-1,0],[2,-1]]];;
gap> BettiNumberToric(Cones,1);
0
gap> BettiNumberToric(Cones,2);
1
```

Not to be confused with the Betti number of a polycyclically presented torsion free group, already available in GAP.

4.3 Points over a finite field

4.3.1 CardinalityOfToricVariety

▷ `CardinalityOfToricVariety(Cones, q)` (function)

Input: `Cones` is the list of maximal cones of a fan Δ, q is a prime power.
Output: The size of the set of $GF(q)$-rational points of the toric variety $X(\Delta)$.

Note: $X(\Delta)$ *must be non-singular* here.

Example

```
gap> Cones:=[[[2,-1],[-1,2]],[[-1,2],[-1,-1]],[[-1,-1],[2,-1]]];;
gap> CardinalityOfToricVariety(Cones,3);
13
gap> CardinalityOfToricVariety(Cones,4);
21
gap> CardinalityOfToricVariety(Cones,5);
31
gap> CardinalityOfToricVariety(Cones,7);
57
```

References

[Ful93] W. Fulton. *Introduction to toric varieties*. Princeton University Press, 1993. 4, 6, 7, 8, 14

[JFM] Cen Tjhai R. Miller T. Boothby R. Baart J. Cramwinckel E. Roijackers J. Fields, D. Joyner and E. Minkes. 4

[JV02] D. Joyner and H. Verrill. *Notes on toric varieties*, posted at math arxiv, http://front.math.ucdavis.edu/math.ag/0208065. *(appeared as Computing with toric varieties, Journal of Symbolic Computation, Volume 42, Issue 5, May 2007, Pages 511-532)*, 2002. 4, 6, 13

Index