ToricVarieties
A package to handle toric varieties
2018.10.12

12 October 2018

Sebastian Gutsche

Martin Bies

Sebastian Gutsche
Email: sebastian.gutsche@rwth-aachen.de
Homepage: http://wwwb.math.rwth-aachen.de/~gutsche/
Address: Sebastian Gutsche
   Lehrstuhl B fuer Mathematik, RWTH Aachen
   Templergraben 64
   52062 Aachen
   Germany

Martin Bies
Email: m.bies@thphys.uni-heidelberg.de
Address: Martin Bies
   Philosophenweg 19
   69120 Heidelberg
   Germany
# Contents

1 Introduction ........................................ 4  
   1.1 What is the goal of the ToricVarieties package? ........ 4

2 Installation of the ToricVarieties Package ................. 5

3 Toric Varieties ........................................ 6  
   3.1 Toric Varieties: Examples .......................... 6  
   3.2 Toric variety: Category and Representations .......... 7  
   3.3 Properties ........................................ 8  
   3.4 Attributes ....................................... 9  
   3.5 Methods ....................................... 12  
   3.6 Constructors ................................... 14

4 Toric subvarieties ....................................... 15  
   4.1 The GAP category ................................ 15  
   4.2 Properties ...................................... 15  
   4.3 Attributes ..................................... 16  
   4.4 Methods ....................................... 16  
   4.5 Constructors ................................... 16

5 Affine toric varieties .................................... 17  
   5.1 Affine toric varieties: Examples .................... 17  
   5.2 The GAP category ................................ 17  
   5.3 Attributes ..................................... 18  
   5.4 Methods ....................................... 18  
   5.5 Constructors ................................... 18

6 Projective toric varieties .................................. 19  
   6.1 Projective toric varieties: Examples ................. 19  
   6.2 The GAP category ................................ 20  
   6.3 Attributes ..................................... 21  
   6.4 Properties ...................................... 21  
   6.5 Methods ....................................... 21  
   6.6 Constructors ................................... 22
7 Toric morphisms
  7.1 Toric morphisms: Examples .................................................. 23
  7.2 The GAP category .............................................................. 24
  7.3 Properties ................................................................. 24
  7.4 Attributes ................................................................. 24
  7.5 Methods ................................................................. 26
  7.6 Constructors ............................................................. 26

8 Toric divisors
  8.1 Toric divisors: Examples .................................................. 27
  8.2 The GAP category .............................................................. 28
  8.3 Properties ................................................................. 28
  8.4 Attributes ................................................................. 29
  8.5 Methods ................................................................. 31
  8.6 Constructors ............................................................. 33

Index 34
Chapter 1

Introduction

1.1 What is the goal of the ToricVarieties package?

ToricVarieties provides data structures to handle toric varieties by their commutative algebra structure and by their combinatorics. For combinatorics, it uses the Convex package. Its goal is to provide a suitable framework to work with toric varieties. All combinatorial structures mentioned in this manual are the ones from Convex.
Chapter 2

Installation of the ToricVarieties Package

• To install this package just extract the package’s archive file to the GAP pkg directory.

• By default the ToricVarieties package is not automatically loaded by GAP when it is installed. You must load the package with the following command, before its functions become available: LoadPackage("ToricVarieties");

• Please, send me an e-mail if you have any questions, remarks, suggestions, etc. concerning this package. Also, I would be please to hear about applications of this package and about any suggestions for new methods to add to the package.

Sebastian Gutsche
Chapter 3

Toric Varieties

3.1 Toric Varieties: Examples

3.1.1 The Hirzebruch surface of index 5

Example

```
gap> H5 := Fan( [[-1,5],[0,1],[1,0],[0,-1]],[[1,2],[2,3],[3,4],[4,1]] );
gap> H5 := ToricVariety( H5 );
gap> IsComplete( H5 );
true
gap> IsAffine( H5 );
false
gap> IsOrbifold( H5 );
true
gap> IsProjective( H5 );
true
gap> TorusInvariantPrimeDivisors( H5 );
[ <A prime divisor of a toric variety with coordinates ( 1, 0, 0, 0 )>,
  <A prime divisor of a toric variety with coordinates ( 0, 1, 0, 0 )>,
  <A prime divisor of a toric variety with coordinates ( 0, 0, 1, 0 )>,
  <A prime divisor of a toric variety with coordinates ( 0, 0, 0, 1 )> ]
gap> P := TorusInvariantPrimeDivisors( H5 );
[ <A prime divisor of a toric variety with coordinates ( 1, 0, 0, 0 )>,
  <A prime divisor of a toric variety with coordinates ( 0, 1, 0, 0 )>,
  <A prime divisor of a toric variety with coordinates ( 0, 0, 1, 0 )>,
  <A prime divisor of a toric variety with coordinates ( 0, 0, 0, 1 )> ]
<A divisor of a toric variety with coordinates ( 1, -1, 4, 0 )>
gap> A;
<A divisor of a toric variety with coordinates ( 1, -1, 4, 0 )>
gap> IsAmple( A );
false
gap> CoordinateRingOfTorus( H5,"x" );
Q[x1,x1_,x2,x2_]/( x1*x1_-1, x2*x2_-1 )
gap> D:=CreateDivisor( [ 0,0,0,0 ],H5 );
<A divisor of a toric variety with coordinates 0>
gap> BasisOfGlobalSections( D );
[ [ 1 ] ]
```
3.1.2 A smooth, complete toric variety which is not projective

Example

\begin{verbatim}
gap> rays := [ [1,0,0], [-1,0,0], [0,1,0], [0,-1,0], [0,0,1], [0,0,-1],
            [2,1,1], [1,2,1], [1,1,2], [1,1,1] ];
gap> cones := [ [1,3,6], [1,4,6], [1,4,5], [2,3,6], [2,4,6], [2,3,5], [2,4,5],
              [1,5,9], [3,5,8], [1,3,7], [1,7,9], [5,8,9], [3,7,8],
              [7,9,10], [8,9,10], [7,8,10] ];
gap> F := Fan( rays, cones );
gap> T := ToricVariety( F );
< A toric variety of dimension 3 >
gap> [ IsSmooth( T ), IsComplete( T ), IsProjective( T ) ];
[ true, true, false ]
gap> SRIdeal( T );
< A graded torsion-free (left) ideal given by 23 generators >
\end{verbatim}

3.2 Toric variety: Category and Representations

3.2.1 IsToricVariety (for IsObject)

\begin{verbatim}
▷ IsToricVariety( M )

Returns: true or false
Checks if an object is a toric variety.
\end{verbatim}

3.2.2 IsCategoryOfToricVarieties (for IsHomalgCategory)

\begin{verbatim}
▷ IsCategoryOfToricVarieties( object )

Returns: true or false
The GAP category of toric varieties.
\end{verbatim}
3.2.3 twitter (for IsToricVariety)

▷ twitter(vari) (attribute)
   Returns: a ring
   This is a dummy to get immediate methods triggered at some times. It never has a value.

3.3 Properties

3.3.1 IsNormalVariety (for IsToricVariety)

▷ IsNormalVariety(vari) (property)
   Returns: true or false
   Checks if the toric variety vari is a normal variety.

3.3.2 IsAffine (for IsToricVariety)

▷ IsAffine(vari) (property)
   Returns: true or false
   Checks if the toric variety vari is an affine variety.

3.3.3 IsProjective (for IsToricVariety)

▷ IsProjective(vari) (property)
   Returns: true or false
   Checks if the toric variety vari is a projective variety.

3.3.4 IsSmooth (for IsToricVariety)

▷ IsSmooth(vari) (property)
   Returns: true or false
   Checks if the toric variety vari is smooth.

3.3.5 IsComplete (for IsToricVariety)

▷ IsComplete(vari) (property)
   Returns: true or false
   Checks if the toric variety vari is complete.

3.3.6 HasTorusfactor (for IsToricVariety)

▷ HasTorusfactor(vari) (property)
   Returns: true or false
   Checks if the toric variety vari has a torus factor.

3.3.7 HasNoTorusfactor (for IsToricVariety)

▷ HasNoTorusfactor(vari) (property)
   Returns: true or false
   Checks if the toric variety vari has no torus factor.
3.3.8 IsOrbifold (for IsToricVariety)

\[
\text{IsOrbifold(vari)} \quad \text{(property)}
\]

**Returns:** true or false

Checks if the toric variety vari has an orbifold, which is, in the toric case, equivalent to the simpliciality of the fan.

3.4 Attributes

3.4.1 AffineOpenCovering (for IsToricVariety)

\[
\text{AffineOpenCovering(vari)} \quad \text{(attribute)}
\]

**Returns:** a list

Returns a torus invariant affine open covering of the variety vari. The affine open cover is computed out of the cones of the fan.

3.4.2 CoxRing (for IsToricVariety)

\[
\text{CoxRing(vari)} \quad \text{(attribute)}
\]

**Returns:** a ring

Returns the Cox ring of the variety vari. The actual method requires a string with a name for the variables. A method for computing the Cox ring without a variable given is not implemented. You will get an error.

3.4.3 ListOfVariablesOfCoxRing (for IsToricVariety)

\[
\text{ListOfVariablesOfCoxRing(vari)} \quad \text{(attribute)}
\]

**Returns:** a list

Returns a list of the variables of the cox ring of the variety vari.

3.4.4 ClassGroup (for IsToricVariety)

\[
\text{ClassGroup(vari)} \quad \text{(attribute)}
\]

**Returns:** a module

Returns the class group of the variety vari as factor of a free module.

3.4.5 TorusInvariantDivisorGroup (for IsToricVariety)

\[
\text{TorusInvariantDivisorGroup(vari)} \quad \text{(attribute)}
\]

**Returns:** a module

Returns the subgroup of the Weil divisor group of the variety vari generated by the torus invariant prime divisors. This is always a finitely generated free module over the integers.

3.4.6 MapFromCharacterToPrincipalDivisor (for IsToricVariety)

\[
\text{MapFromCharacterToPrincipalDivisor(vari)} \quad \text{(attribute)}
\]

**Returns:** a morphism

Returns a map which maps an element of the character group into the torus invariant Weil group of the variety vari. This has to be viewed as a help method to compute divisor classes.
3.4.7 MapFromWeilDivisorsToClassGroup (for IsToricVariety)

- MapFromWeilDivisorsToClassGroup(vari) (attribute)
  - **Returns**: a morphism
    Returns a map which maps a Weil divisor into the class group.

3.4.8 Dimension (for IsToricVariety)

- Dimension(vari) (attribute)
  - **Returns**: an integer
    Returns the dimension of the variety vari.

3.4.9 DimensionOfTorusfactor (for IsToricVariety)

- DimensionOfTorusfactor(vari) (attribute)
  - **Returns**: an integer
    Returns the dimension of the torus factor of the variety vari.

3.4.10 CoordinateRingOfTorus (for IsToricVariety)

- CoordinateRingOfTorus(vari) (attribute)
  - **Returns**: a ring
    Returns the coordinate ring of the torus of the variety vari. This is by default done with the variables x1 to xn where n is the dimension of the variety. To use a different set of variables, a convenience method is provided and described in the methods section.

3.4.11 ListOfVariablesOfCoordinateRingOfTorus (for IsToricVariety)

- ListOfVariablesOfCoordinateRingOfTorus(vari) (attribute)
  - **Returns**: a list
    Returns the list of variables in the coordinate ring of the torus of the variety vari.

3.4.12 IsProductOf (for IsToricVariety)

- IsProductOf(vari) (attribute)
  - **Returns**: a list
    If the variety vari is a product of 2 or more varieties, the list contains those varieties. If it is not a product or at least not generated as a product, the list only contains the variety itself.

3.4.13 CharacterLattice (for IsToricVariety)

- CharacterLattice(vari) (attribute)
  - **Returns**: a module
    The method returns the character lattice of the variety vari, computed as the containing grid of the underlying convex object, if it exists.
3.4.14 TorusInvariantPrimeDivisors (for IsToricVariety)

\[ \text{TorusInvariantPrimeDivisors(vari)} \]

\text{(attribute)}

\text{Returns: a list}

The method returns a list of the torus invariant prime divisors of the variety vari.

3.4.15 IrrelevantIdeal (for IsToricVariety)

\[ \text{IrrelevantIdeal(vari)} \]

\text{(attribute)}

\text{Returns: an ideal}

Returns the irrelevant ideal of the Cox ring of the variety vari.

3.4.16 SRIdeal (for IsToricVariety)

\[ \text{SRIdeal(vari)} \]

\text{(attribute)}

\text{Returns: an ideal}

Returns the Stanley-Reißner ideal of the Cox ring of the variety vari.

3.4.17 MorphismFromCoxVariety (for IsToricVariety)

\[ \text{MorphismFromCoxVariety(vari)} \]

\text{(attribute)}

\text{Returns: a morphism}

The method returns the quotient morphism from the variety of the Cox ring to the variety vari.

3.4.18 CoxVariety (for IsToricVariety)

\[ \text{CoxVariety(vari)} \]

\text{(attribute)}

\text{Returns: a variety}

The method returns the Cox variety of the variety vari.

3.4.19 FanOfVariety (for IsToricVariety)

\[ \text{FanOfVariety(vari)} \]

\text{(attribute)}

\text{Returns: a fan}

Returns the fan of the variety vari. This is set by default.

3.4.20 CartierTorusInvariantDivisorGroup (for IsToricVariety)

\[ \text{CartierTorusInvariantDivisorGroup(vari)} \]

\text{(attribute)}

\text{Returns: a module}

Returns the the group of Cartier divisors of the variety vari as a subgroup of the divisor group.

3.4.21 PicardGroup (for IsToricVariety)

\[ \text{PicardGroup(vari)} \]

\text{(attribute)}

\text{Returns: a module}

Returns the Picard group of the variety vari as factor of a free module.
3.4.22 NameOfVariety (for IsToricVariety)

- NameOfVariety(vari)
  - Attributes: a string
  - Returns: a string
  - Returns the name of the variety vari if it has one and it is known or can be computed.

3.4.23 ZariskiCotangentSheaf (for IsToricVariety)

- ZariskiCotangentSheaf(vari)
  - Attributes: a f.p. graded S-module
  - Returns: a f.p. graded S-module
  - This method returns a f. p. graded S-module (S being the Cox ring of the variety), such that the sheafification of this module is the Zariski cotangent sheaf of vari.

3.4.24 CotangentSheaf (for IsToricVariety)

- CotangentSheaf(vari)
  - Attributes: a f.p. graded S-module
  - Returns: a f.p. graded S-module
  - This method returns a f. p. graded S-module (S being the Cox ring of the variety), such that the sheafification of this module is the cotangent sheaf of vari.

3.4.25 EulerCharacteristic (for IsToricVariety)

- EulerCharacteristic(vari)
  - Attributes: a non-negative integer
  - Returns: a non-negative integer
  - This method computes the Euler characteristic of vari.

3.5 Methods

3.5.1 UnderlyingSheaf (for IsToricVariety)

- UnderlyingSheaf(vari)
  - Attributes: a sheaf
  - Returns: a sheaf
  - The method returns the underlying sheaf of the variety vari.

3.5.2 CoordinateRingOfTorus (for IsToricVariety, IsList)

- CoordinateRingOfTorus(vari, vars)
  - Attributes: a ring
  - Returns: a ring
  - Computes the coordinate ring of the torus of the variety vari with the variables vars. The argument vars need to be a list of strings with length dimension or two times dimension.

3.5.3 \* (for IsToricVariety, IsToricVariety)

- \*(vari1, vari2)
  - Attributes: a variety
  - Returns: a variety
  - Computes the categorial product of the varieties vari1 and vari2.
3.5.4  CharacterToRationalFunction (for IsHomalgElement, IsToricVariety)

> CharacterToRationalFunction(elem, vari)  
   (operation)

   Returns: a homalg element

   Computes the rational function corresponding to the character grid element \( \text{elem} \) or to the list of integers \( \text{elem} \). This computation needs to know the coordinate ring of the torus of the variety \( \text{vari} \). By default this ring is introduced with variables \( x_1 \) to \( x_n \) where \( n \) is the dimension of the variety. If different variables should be used, then \text{CoordinateRingOfTorus} \ has to be set accordingly before calling this method.

3.5.5  CoxRing (for IsToricVariety, IsString)

> CoxRing(vari, vars)  
   (operation)

   Returns: a ring

   Computes the Cox ring of the variety \( \text{vari} \). \( \text{vars} \) needs to be a string containing one variable, which will be numbered by the method.

3.5.6  WeilDivisorsOfVariety (for IsToricVariety)

> WeilDivisorsOfVariety(vari)  
   (operation)

   Returns: a list

   Returns a list of the currently defined Divisors of the toric variety.

3.5.7  Fan (for IsToricVariety)

> Fan(vari)  
   (operation)

   Returns: a fan

   Returns the fan of the variety \( \text{vari} \). This is a rename for \text{FanOfVariety}.

3.5.8  Factors (for IsToricVariety)

> Factors(vari)  
   (operation)

3.5.9  BlowUpOnIthMinimalTorusOrbit (for IsToricVariety, IsInt)

> BlowUpOnIthMinimalTorusOrbit(vari, p)  
   (operation)

3.5.10 ZariskiCotangentSheafViaEulerSequence

> ZariskiCotangentSheafViaEulerSequence(arg)  
   (function)

3.5.11 ZariskiCotangentSheafViaPoincareResidueMap

> ZariskiCotangentSheafViaPoincareResidueMap(arg)  
   (function)
3.5.12 \textbf{ithBettiNumber} (for \texttt{IsToric Variety}, \texttt{IsInt})

\[
\texttt{ithBettiNumber(vari, p)} \quad \text{(operation)}
\]

3.5.13 \textbf{NrOfqRationalPoints} (for \texttt{IsToric Variety}, \texttt{IsInt})

\[
\texttt{NrOfqRationalPoints(vari, p)} \quad \text{(operation)}
\]

3.6 \textbf{Constructors}

3.6.1 \textbf{Toric Variety} (for \texttt{IsToric Variety})

\[
\texttt{Toric Variety(vari)} \quad \text{(operation)}
\]

3.6.2 \textbf{Toric Variety} (for \texttt{IsList})

\[
\texttt{Toric Variety(vari)} \quad \text{(operation)}
\]

3.6.3 \textbf{Toric Variety} (for \texttt{IsConvexObject})

\[
\texttt{Toric Variety(conv)} \quad \text{(operation)}
\]

\textbf{Returns}: a variety

Creates a toric variety out of the convex object \textit{conv}.

3.6.4 \textbf{Toric Variety} (for \texttt{IsConvexObject}, \texttt{IsList})

\[
\texttt{Toric Variety(conv)} \quad \text{(operation)}
\]

\textbf{Returns}: a variety

Creates a toric variety out of the convex object \textit{conv}. In addition it takes a list of integers. Those are used to set the degrees of the variables in the Cox ring of the variety.
Chapter 4

Toric subvarieties

4.1 The GAP category

4.1.1 IsToricSubvariety (for IsToricVariety)

\[ \text{IsToricSubvariety}(M) \]

*Returns:* true or false

The GAP category of a toric subvariety. Every toric subvariety is a toric variety, so every method applicable to toric varieties is also applicable to toric subvarieties.

4.2 Properties

4.2.1 IsClosedSubvariety (for IsToricSubvariety)

\[ \text{IsClosedSubvariety}(\text{vari}) \]

*Returns:* true or false

Checks if the subvariety \text{vari} is a closed subset of its ambient variety.

4.2.2 IsOpen (for IsToricSubvariety)

\[ \text{IsOpen}(\text{vari}) \]

*Returns:* true or false

Checks if a subvariety is a closed subset.

4.2.3 IsWholeVariety (for IsToricSubvariety)

\[ \text{IsWholeVariety}(\text{vari}) \]

*Returns:* true or false

Returns true if the subvariety \text{vari} is the whole variety.
4.3 Attributes

4.3.1 UnderlyingToricVariety (for IsToricSubvariety)

\[ \text{UnderlyingToricVariety(vari)} \]

**Returns:** a variety

Returns the toric variety which is represented by vari. This method implements the forgetful functor subvarieties \( \rightarrow \) varieties.

4.3.2 InclusionMorphism (for IsToricSubvariety)

\[ \text{InclusionMorphism(vari)} \]

**Returns:** a morphism

If the variety vari is an open subvariety, this method returns the inclusion morphism in its ambient variety. If not, it will fail.

4.3.3 AmbientToricVariety (for IsToricSubvariety)

\[ \text{AmbientToricVariety(vari)} \]

**Returns:** a variety

Returns the ambient toric variety of the subvariety vari

4.4 Methods

4.4.1 ClosureOfTorusOrbitOfCone (for IsToricVariety, IsCone)

\[ \text{ClosureOfTorusOrbitOfCone(vari, cone)} \]

**Returns:** a subvariety

The method returns the closure of the orbit of the torus contained in vari which corresponds to the cone cone as a closed subvariety of vari.

4.5 Constructors

4.5.1 ToricSubvariety (for IsToricVariety, IsToricVariety)

\[ \text{ToricSubvariety(vari, ambvari)} \]

**Returns:** a subvariety

The method returns the closure of the orbit of the torus contained in vari which corresponds to the cone cone as a closed subvariety of vari.
Chapter 5

Affine toric varieties

5.1 Affine toric varieties: Examples

5.1.1 Affine space

Example

\[
gap> C:=\text{Cone}( \begin{bmatrix} 1,0,0 \end{bmatrix}, \begin{bmatrix} 0,1,0 \end{bmatrix}, \begin{bmatrix} 0,0,1 \end{bmatrix} );
\]
\[
\text{<A cone in } \mathbb{R}^3 \text{>}
\]
\[
gap> C3:=\text{ToricVariety}(C);
\]
\[
\text{<An affine normal toric variety of dimension 3>}
\]
\[
gap> \text{Dimension}(C3);
\]
\[
3
\]
\[
gap> \text{IsOrbifold}(C3);
\]
\[
\text{true}
\]
\[
gap> \text{IsSmooth}(C3);
\]
\[
\text{true}
\]
\[
gap> \text{CoordinateRingOfTorus}(C3, "x");
\]
\[
\mathbb{Q}[x1,x1_,x2,x2_,x3,x3_]/(x1*x1_-1, x2*x2_-1, x3*x3_-1)
\]
\[
gap> \text{CoordinateRing}(C3, "x");
\]
\[
\mathbb{Q}[x_1,x_2,x_3]
\]
\[
gap> \text{MorphismFromCoordinateRingToCoordinateRingOfTorus}(C3);
\]
\[
\text{<A monomorphism of rings>}
\]
\[
gap> C3;
\]
\[
\text{<An affine normal smooth toric variety of dimension 3>}
\]
\[
gap> \text{StructureDescription}(C3);
\]
\[
"|A^-3$
\]

5.2 The GAP category

5.2.1 IsAffineToricVariety (for IsToricVariety)

\[
\text{IsAffineToricVariety}(M) \quad \text{(filter)}
\]

Returns: true or false

The GAP category of an affine toric variety. All affine toric varieties are toric varieties, so everything applicable to toric varieties is applicable to affine toric varieties.
5.3 Attributes

5.3.1 CoordinateRing (for IsAffineToricVariety)

\[ \text{CoordinateRing}(\text{vari}) \]
\[ \text{attribute} \]

Returns: a ring

Returns the coordinate ring of the affine toric variety \( \text{vari} \). The computation is mainly done in ToricIdeals package.

5.3.2 ListOfVariablesOfCoordinateRing (for IsAffineToricVariety)

\[ \text{ListOfVariablesOfCoordinateRing}(\text{vari}) \]
\[ \text{attribute} \]

Returns: a list

Returns a list containing the variables of the CoordinateRing of the variety \( \text{vari} \).

5.3.3 MorphismFromCoordinateRingToCoordinateRingOfTorus (for IsToricVariety)

\[ \text{MorphismFromCoordinateRingToCoordinateRingOfTorus}(\text{vari}) \]
\[ \text{attribute} \]

Returns: a morphism

Returns the morphism between the coordinate ring of the variety \( \text{vari} \) and the coordinate ring of its torus. This defines the embedding of the torus in the variety.

5.3.4 ConeOfVariety (for IsToricVariety)

\[ \text{ConeOfVariety}(\text{vari}) \]
\[ \text{attribute} \]

Returns: a cone

Returns the cone of the affine toric variety \( \text{vari} \).

5.4 Methods

5.4.1 CoordinateRing (for IsToricVariety, IsList)

\[ \text{CoordinateRing}(\text{vari}, \text{indet}) \]
\[ \text{operation} \]

Returns: a ring

Computes the coordinate ring of the affine toric variety \( \text{vari} \) with indeterminates \( \text{indet} \).

5.4.2 Cone (for IsToricVariety)

\[ \text{Cone}(\text{vari}) \]
\[ \text{operation} \]

Returns: a cone

Returns the cone of the variety \( \text{vari} \). Another name for ConeOfVariety for compatibility and shortness.

5.5 Constructors

The constructors are the same as for toric varieties. Calling them with a cone will result in an affine variety.
Chapter 6

Projective toric varieties

6.1 Projective toric varieties: Examples

6.1.1 P1xP1 created by a polytope

Example

```gap
gap> P1P1 := Polytope( [[1,1],[1,-1],[-1,-1],[-1,1]] );
< A polytope in |R^2>
gap> P1P1 := ToricVariety( P1P1 );
< A projective toric variety of dimension 2>
gap> IsProjective( P1P1 );
true
gap> IsComplete( P1P1 );
true
gap> CoordinateRingOfTorus( P1P1, "x" );
Q[x1,x1_,x2,x2_]/( x1*x1_-1, x2*x2_-1 )
gap> IsVeryAmple( Polytope( P1P1 ) );
true

gap> ProjectiveEmbedding( P1P1 );
[ |[ x1_*x2_ ]|, |[ x1_ ]|, |[ x1*x2 ]|, |[ x2_ ]|,
 |[ 1 ]|, |[ x2 ]|, |[ x1*x2_ ]|, |[ x1 ]|, |[ x1*x2 ]| ]

gap> Length( ProjectiveEmbedding( P1P1 ) );
9

gap> CoxRing( P1P1 );
Q[x_1,x_2,x_3,x_4]
(weights: [ ( 0, 1 ), ( 1, 0 ), ( 1, 0 ), ( 0, 1 ) ])

gap> Display( SRIdeal( P1P1 ) );
 x_1*x_4,
x_2*x_3

gap> Display( IrrelevantIdeal( P1P1 ) );
x_1*x_2,
x_1*x_3,
x_2*x_4,
x_3*x_4
```

A (left) ideal generated by the 2 entries of the above matrix

{(graded, degrees of generators: [ ( 0, 2 ), ( 2, 0 ) ])

```gap
```

19
A (left) ideal generated by the 4 entries of the above matrix
(graded, degrees of generators: [ ( 1, 1 ), ( 1, 1 ), ( 1, 1 ), ( 1, 1 ) ])

6.1.2 P1xP1 from product of P1s

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>gap&gt; P1 := ProjectiveSpace( 1 );</td>
</tr>
<tr>
<td>&lt;A projective toric variety of dimension 1&gt;</td>
</tr>
<tr>
<td>gap&gt; IsComplete( P1 );</td>
</tr>
<tr>
<td>true</td>
</tr>
<tr>
<td>gap&gt; IsSmooth( P1 );</td>
</tr>
<tr>
<td>true</td>
</tr>
<tr>
<td>gap&gt; Dimension( P1 );</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>gap&gt; P1xP1 := P1*P1;</td>
</tr>
<tr>
<td>&lt;A projective smooth toric variety of dimension 2 which is a product of 2 toric varieties&gt;</td>
</tr>
<tr>
<td>gap&gt; ByASmallerPresentation( ClassGroup( P1xP1 ) );</td>
</tr>
<tr>
<td>&lt;A free left module of rank 2 on free generators&gt;</td>
</tr>
<tr>
<td>gap&gt; CoxRing( P1xP1 );</td>
</tr>
<tr>
<td>Q[x_1, x_2, x_3, x_4]</td>
</tr>
<tr>
<td>(weights: [ ( 0, 1 ), ( 1, 0 ), ( 1, 0 ), ( 0, 1 ) ])</td>
</tr>
<tr>
<td>gap&gt; Display( SRIdeal( P1xP1 ) );</td>
</tr>
<tr>
<td>x_1*x_4,</td>
</tr>
<tr>
<td>x_2*x_3</td>
</tr>
</tbody>
</table>

A (left) ideal generated by the 2 entries of the above matrix
(graded, degrees of generators: [ ( 0, 2 ), ( 2, 0 ) ])

gap> Display( IrrelevantIdeal( P1xP1 ) ); |
| x_1*x_2, |
| x_1*x_3, |
| x_2*x_4, |
| x_3*x_4 |

A (left) ideal generated by the 4 entries of the above matrix
(graded, degrees of generators: [ ( 1, 1 ), ( 1, 1 ), ( 1, 1 ), ( 1, 1 ) ])

6.2 The GAP category

6.2.1 IsProjectiveToricVariety (for IsToricVariety)

- IsProjectiveToricVariety(M) [(filter)]

  Returns: true or false

  The GAP category of a projective toric variety.
6.3 Attribute

6.3.1 PolytopeOfVariety (for IsToricVariety)

\[
\text{PolytopeOfVariety}(\text{vari})
\]

\textbf{Returns:} a polytope

Returns the polytope corresponding to the projective toric variety vari, if it exists.

6.3.2 AffineCone (for IsToricVariety)

\[
\text{AffineCone}(\text{vari})
\]

\textbf{Returns:} a cone

Returns the affine cone of the projective toric variety vari.

6.3.3 ProjectiveEmbedding (for IsToricVariety)

\[
\text{ProjectiveEmbedding}(\text{vari})
\]

\textbf{Returns:} a list

Returns characters for a closed embedding in an projective space for the projective toric variety vari.

6.4 Properties

6.4.1 IsProjectiveSpace (for IsToricVariety)

\[
\text{IsProjectiveSpace}(\text{vari})
\]

\textbf{Returns:} true or false

Checks if the given toric variety vari is a projective space.

6.4.2 IsDirectProductOfPNs (for IsToricVariety)

\[
\text{IsDirectProductOfPNs}(\text{vari})
\]

\textbf{Returns:} true or false

Checks if the given toric variety vari is a direct product of projective spaces.

6.5 Methods

6.5.1 Polytope (for IsToricVariety)

\[
\text{Polytope}(\text{vari})
\]

\textbf{Returns:} a polytope

Returns the polytope of the variety vari. Another name for PolytopeOfVariety for compatibility and shortness.
6.5.2 AmpleDivisor (for IsToricVariety and HasPolytopeOfVariety)

\[ \text{AmpleDivisor}(\text{vari}) \]

\textbf{Returns:} an ample divisor

Given a projective toric variety \textit{vari} constructed from a polytope, this method computes the toric divisor associated to this polytope. By general theory (see Cox-Schenk-Little) this divisor is known to be ample. Thus this method computes an ample divisor on the given toric variety.

6.6 Constructors

The constructors are the same as for toric varieties. Calling them with a polytope will result in a projective variety.
Chapter 7

Toric morphisms

7.1 Toric morphisms: Examples

7.1.1 Morphism between toric varieties and their class groups

```
gap> P1 := Polytope([[0],[1]]);  # A polytope in \( \mathbb{R}^1 \)
gap> P2 := Polytope([[0,0],[0,1],[1,0]]);  # A polytope in \( \mathbb{R}^2 \)
gap> P1 := ToricVariety( P1 );  # A projective toric variety of dimension 1
<...>
gap> P1P2 := P1*P2;  # A projective toric variety of dimension 3 which is a product of 2 toric varieties
<...>
gap> ClassGroup( P1 );  # A free left module of rank 1 on a free generator
<...>
gap> Display(ByASmallerPresentation(ClassGroup( P1 )));  # \( \mathbb{Z}^{1 \times 1} \)
gap> ClassGroup( P2 );  # A free left module of rank 1 on a free generator
<...>
gap> Display(ByASmallerPresentation(ClassGroup( P2 )));  # \( \mathbb{Z}^{1 \times 1} \)
gap> ClassGroup( P1P2 );  # A free left module of rank 2 on free generators
<...>
gap> Display( last );  # \( \mathbb{Z}^{1 \times 2} \)
gap> PicardGroup( P1P2 );  # A free left module of rank 2 on free generators
<...>
gap> P1P2;  # A projective smooth toric variety of dimension 3 which is a product of 2 toric varieties
<...>
gap> P2P1 := P2*P1;  # A projective toric variety of dimension 3 which is a product of 2 toric varieties
<...>
gap> M := [[0,0,1],[1,0,0],[0,1,0]];  # a matrix
[ [ 0, 0, 1 ], [ 1, 0, 0 ], [ 0, 1, 0 ] ]
```
gap> M := ToricMorphism(P1P2,M,P2P1);
<A "homomorphism" of right objects>
gap> IsMorphism(M);
true
gap> ClassGroup(M);
<A homomorphism of left modules>
gap> Display(ClassGroup(M));
\[
\begin{bmatrix}
  0 & 1 \\
  1 & 0 \\
\end{bmatrix}
\]
the map is currently represented by the above 2 x 2 matrix

7.2 The GAP category

7.2.1 IsToricMorphism (for IsObject)

▷ IsToricMorphism(M)  \hspace{1cm} \text{(filter)}
\hspace{1cm} \text{Returns: true or false}
\hspace{1cm} The GAP category of toric morphisms. A toric morphism is defined by a grid homomorphism, which is compatible with the fan structure of the two varieties.

7.3 Properties

7.3.1 IsMorphism (for IsToricMorphism)

▷ IsMorphism(morph)  \hspace{1cm} \text{(property)}
\hspace{1cm} \text{Returns: true or false}
\hspace{1cm} Checks if the grid morphism \textit{morph} respects the fan structure.

7.3.2 IsProper (for IsToricMorphism)

▷ IsProper(morph)  \hspace{1cm} \text{(property)}
\hspace{1cm} \text{Returns: true or false}
\hspace{1cm} Checks if the defined morphism \textit{morph} is proper.

7.4 Attributes

7.4.1 SourceObject (for IsToricMorphism)

▷ SourceObject(morph)  \hspace{1cm} \text{(attribute)}
\hspace{1cm} \text{Returns: a variety}
\hspace{1cm} Returns the source object of the morphism \textit{morph}. This attribute is a must have.

7.4.2 UnderlyingGridMorphism (for IsToricMorphism)

▷ UnderlyingGridMorphism(morph)  \hspace{1cm} \text{(attribute)}
\hspace{1cm} \text{Returns: a map}
\hspace{1cm} Returns the grid map which defines \textit{morph}. 
7.4.3 ToricImageObject (for IsToricMorphism)

\[ \text{ToricImageObject(\textit{morph})} \]

\textbf{Returns:} a variety

Returns the variety which is created by the fan which is the image of the fan of the source of \textit{morph}. This is not an image in the usual sense, but a toric image.

7.4.4 RangeObject (for IsToricMorphism)

\[ \text{RangeObject(\textit{morph})} \]

\textbf{Returns:} a variety

Returns the range of the morphism \textit{morph}. If no range is given (yes, this is possible), the method returns the image.

7.4.5 MorphismOnWeilDivisorGroup (for IsToricMorphism)

\[ \text{MorphismOnWeilDivisorGroup(\textit{morph})} \]

\textbf{Returns:} a morphism

Returns the associated morphism between the divisor group of the range of \textit{morph} and the divisor group of the source.

7.4.6 ClassGroup (for IsToricMorphism)

\[ \text{ClassGroup(\textit{morph})} \]

\textbf{Returns:} a morphism

Returns the associated morphism between the class groups of source and range of the morphism \textit{morph}.

7.4.7 MorphismOnCartierDivisorGroup (for IsToricMorphism)

\[ \text{MorphismOnCartierDivisorGroup(\textit{morph})} \]

\textbf{Returns:} a morphism

Returns the associated morphism between the Cartier divisor groups of source and range of the morphism \textit{morph}.

7.4.8 PicardGroup (for IsToricMorphism)

\[ \text{PicardGroup(\textit{morph})} \]

\textbf{Returns:} a morphism

Returns the associated morphism between the Picard groups of source and range of the morphism \textit{morph}.

7.4.9 Source (for IsToricMorphism)

\[ \text{Source(\textit{morph})} \]

\textbf{Returns:} a variety

Return the source of the toric morphism \textit{morph}.
7.4.10 Range (for IsToricMorphism)

- Range(morph) (attribute)
  Returns: a variety
  Returns the range of the toric morphism morph if specified.

7.4.11 MorphismOnIthFactor (for IsToricMorphism)

- MorphismOnIthFactor(morph) (attribute)

7.5 Methods

7.5.1 UnderlyingListList (for IsToricMorphism)

- UnderlyingListList(morph) (operation)
  Returns: a list
  Returns a list of list which represents the grid homomorphism.

7.6 Constructors

7.6.1 ToricMorphism (for IsToricVariety, IsList)

- ToricMorphism(vari, lis) (operation)
  Returns: a morphism
  Returns the toric morphism with source vari which is represented by the matrix lis. The range is set to the image.

7.6.2 ToricMorphism (for IsToricVariety, IsList, IsToricVariety)

- ToricMorphism(vari, lis, vari2) (operation)
  Returns: a morphism
  Returns the toric morphism with source vari and range vari2 which is represented by the matrix lis.
Chapter 8

Toric divisors

8.1 Toric divisors: Examples

8.1.1 Divisors on a toric variety

```gap
gap> H7 := Fan( [[0,1],[1,0],[0,-1],[-1,7]],[[1,2],[2,3],[3,4],[4,1]] );
<A fan in |R^2>
gap> H7 := ToricVariety( H7 );
<A toric variety of dimension 2>
gap> P := TorusInvariantPrimeDivisors( H7 );
[ <A prime divisor of a toric variety with coordinates ( 1, 0, 0, 0 )>,
  <A prime divisor of a toric variety with coordinates ( 0, 1, 0, 0 )>,
  <A prime divisor of a toric variety with coordinates ( 0, 0, 1, 0 )>,
  <A prime divisor of a toric variety with coordinates ( 0, 0, 0, 1 )> ]
gap> D := P[1]+P[2];
<A divisor of a toric variety with coordinates ( 1, 1, 0, 0 )>
gap> IsBasepointFree(D);
true
gap> IsAmple(D);
true
```

```gap
gap> CoordinateRingOfTorus(H7,"x");
Q[x1,x1_,x2,x2_]/( x1*x1_-1, x2*x2_-1 )
gap> Polytope(D);
<A polytope in |R^2>
```

```gap
gap> CharactersForClosedEmbedding(D);
[ [ 1 ], [ 1 ], [ 1 ], [ 1 ], [ 1 ], [ 1 ], [ 1 ], [ 1 ], [ 1 ], [ 1 ], [ 1 ] ]
gap> CoxRingOfTargetOfDivisorMorphism(D);
Q[x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_10,x_11]
(weights: [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ])
gap> RingMorphismOfDivisor(D);
<A "homomorphism" of rings>
```

```gap
gap> Display(RingMorphismOfDivisor(D));
Q[x_1,x_2,x_3,x_4]
(weights: [ ( 0, 1 ), ( 1, 0 ), ( 1, -7 ), ( 0, 1 ) ])
```
```
[ \ x_1*x_2, \ x_1^8*x_3, \ x_2*x_4, \ x_1^7*x_3*x_4, \ x_1^6*x_3*x_4^2, \ x_1^5*x_3*x_4^3, \ x_1^4*x_3*x_4^4, \ x_1^3*x_3*x_4^5, \ x_1^2*x_3*x_4^6, \ x_1*x_3*x_4^7, \ x_3*x_4^8 \ ]

| Q[x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_10,x_11] (weights: [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ])

\texttt{gap} \rightarrow \texttt{ByASmallerPresentation(ClassGroup(H7));}
\texttt{<A free left module of rank 2 on free generators>}
\texttt{gap} \rightarrow \texttt{MonomsOfCoxRingOfDegree(D)};
\texttt{[ \ x_1*x_2, \ x_1^8*x_3, \ x_2*x_4, \ x_1^7*x_3*x_4, \ x_1^6*x_3*x_4^2, \ x_1^5*x_3*x_4^3, \ x_1^4*x_3*x_4^4, \ x_1^3*x_3*x_4^5, \ x_1^2*x_3*x_4^6, \ x_1*x_3*x_4^7, \ x_3*x_4^8 \ ]}
\texttt{gap} \rightarrow \texttt{D2:=D-2*P[2];}
\texttt{<A divisor of a toric variety with coordinates ( 1, -1, 0, 0 )>}
\texttt{gap} \rightarrow \texttt{IsBasepointFree(D2);}
\texttt{false}
\texttt{gap} \rightarrow \texttt{IsAmple(D2);}
\texttt{false}

8.2 The \textit{GAP} category

8.2.1 \texttt{IsToricDivisor} (for \texttt{IsObject})

\texttt{IsToricDivisor}(M) \hspace{1cm} \text{(filter)}
\textbf{Returns:} true or false
The \textit{GAP} category of torus invariant Weil divisors.

8.2.2 \texttt{twitter} (for \texttt{IsToricDivisor})

\texttt{twitter(arg)} \hspace{1cm} \text{(attribute)}

8.3 Properties

8.3.1 \texttt{IsCartier} (for \texttt{IsToricDivisor})

\texttt{IsCartier(divi)} \hspace{1cm} \text{(property)}
\textbf{Returns:} true or false
Checks if the torus invariant Weil divisor \texttt{divi} is Cartier i.e. if it is locally principal.

8.3.2 \texttt{IsPrincipal} (for \texttt{IsToricDivisor})

\texttt{IsPrincipal(divi)} \hspace{1cm} \text{(property)}
\textbf{Returns:} true or false
Checks if the torus invariant Weil divisor \texttt{divi} is principal which in the toric invariant case means that it is the divisor of a character.
8.3.3 IsPrimedivisor (for IsToricDivisor)

\[ \text{IsPrimedivisor}(\text{divi}) \]  
\textbf{Returns:} true or false  
Checks if the Weil divisor \( \text{divi} \) represents a prime divisor, i.e. if it is a standard generator of the divisor group.

8.3.4 IsBasepointFree (for IsToricDivisor)

\[ \text{IsBasepointFree}(\text{divi}) \]  
\textbf{Returns:} true or false  
Checks if the divisor \( \text{divi} \) is basepoint free.

8.3.5 IsAmple (for IsToricDivisor)

\[ \text{IsAmple}(\text{divi}) \]  
\textbf{Returns:} true or false  
Checks if the divisor \( \text{divi} \) is ample, i.e. if it is colored red, yellow and green.

8.3.6 IsVeryAmple (for IsToricDivisor)

\[ \text{IsVeryAmple}(\text{divi}) \]  
\textbf{Returns:} true or false  
Checks if the divisor \( \text{divi} \) is very ample.

8.3.7 IsNumericallyEffective (for IsToricDivisor)

\[ \text{IsNumericallyEffective}(\text{divi}) \]  
\textbf{Returns:} true or false  
Checks if the divisor \( \text{divi} \) is nef.

8.4 Attributes

8.4.1 CartierData (for IsToricDivisor)

\[ \text{CartierData}(\text{divi}) \]  
\textbf{Returns:} a list  
Returns the Cartier data of the divisor \( \text{divi} \), if it is Cartier, and fails otherwise.

8.4.2 CharacterOfPrincipalDivisor (for IsToricDivisor)

\[ \text{CharacterOfPrincipalDivisor}(\text{divi}) \]  
\textbf{Returns:} a homalg module element  
Returns the character corresponding to the principal divisor \( \text{divi} \).
8.4.3 ClassOfDivisor (for IsToricDivisor)

▷ ClassOfDivisor(divi) (attribute)
  Returns: a homalg module element
  Returns the class group element corresponding to the divisor divi.

8.4.4 PolytopeOfDivisor (for IsToricDivisor)

▷ PolytopeOfDivisor(divi) (attribute)
  Returns: a polytope
  Returns the polytope corresponding to the divisor divi.

8.4.5 BasisOfGlobalSections (for IsToricDivisor)

▷ BasisOfGlobalSections(divi) (attribute)
  Returns: a list
  Returns a basis of the global section module of the quasi-coherent sheaf of the divisor divi.

8.4.6 IntegerForWhichIsSureVeryAmple (for IsToricDivisor)

▷ IntegerForWhichIsSureVeryAmple(divi) (attribute)
  Returns: an integer
  Returns an integer n such that n · divi is very ample.

8.4.7 AmbientToricVariety (for IsToricDivisor)

▷ AmbientToricVariety(divi) (attribute)
  Returns: a variety
  Returns the toric variety which contains the prime divisors of the divisor divi.

8.4.8 UnderlyingGroupElement (for IsToricDivisor)

▷ UnderlyingGroupElement(divi) (attribute)
  Returns: a homalg module element
  Returns an element which represents the divisor divi in the Weil group.

8.4.9 UnderlyingToricVariety (for IsToricDivisor)

▷ UnderlyingToricVariety(divi) (attribute)
  Returns: a variety
  Returns the closure of the torus orbit corresponding to the prime divisor divi. Not implemented for other divisors. Maybe we should add the support here. Is this even a toric variety? Exercise left to the reader.
8.4.10 DegreeOfDivisor (for IsToricDivisor)

▶ DegreeOfDivisor(divi) (attribute)
  Returns: an integer
  Returns the degree of the divisor \( divi \). This is not to be confused with the (divisor) class of \( divi \).

8.4.11 VarietyOfDivisorpolytope (for IsToricDivisor)

▶ VarietyOfDivisorpolytope(divi) (attribute)
  Returns: a variety
  Returns the variety corresponding to the polytope of the divisor \( divi \).

8.4.12 MonomsOfCoxRingOfDegree (for IsToricDivisor)

▶ MonomsOfCoxRingOfDegree(divi) (attribute)
  Returns: a list
  Returns the monoms in the Cox ring of degree equal to the (divisor) class of the divisor \( divi \).

8.4.13 CoxRingOfTargetOfDivisorMorphism (for IsToricDivisor)

▶ CoxRingOfTargetOfDivisorMorphism(divi) (attribute)
  Returns: a ring
  A basepoint free divisor \( divi \) defines a map from its ambient variety in a projective space. This method returns the Cox ring of such a projective space.

8.4.14 RingMorphismOfDivisor (for IsToricDivisor)

▶ RingMorphismOfDivisor(divi) (attribute)
  Returns: a ring map
  A basepoint free divisor \( divi \) defines a map from its ambient variety in a projective space. This method returns the morphism between the cox ring of this projective space to the cox ring of the ambient variety of \( divi \).

8.5 Methods

8.5.1 VeryAmpleMultiple (for IsToricDivisor)

▶ VeryAmpleMultiple(divi) (operation)
  Returns: a divisor
  Returns a very ample multiple of the ample divisor \( divi \). The method will fail if divisor is not ample.

8.5.2 CharactersForClosedEmbedding (for IsToricDivisor)

▶ CharactersForClosedEmbedding(divi) (operation)
  Returns: a list
  Returns characters for closed embedding defined via the ample divisor \( divi \). The method fails if the divisor \( divi \) is not ample.
8.5.3 \( + \) (for IsToricDivisor, IsToricDivisor)

- \( +(\text{divi}_1, \text{divi}_2) \) (operation)
  - \textbf{Returns:} a divisor
  - Returns the sum of the divisors \text{divi}_1 and \text{divi}_2.

8.5.4 \( - \) (for IsToricDivisor, IsToricDivisor)

- \( -(\text{divi}_1, \text{divi}_2) \) (operation)
  - \textbf{Returns:} a divisor
  - Returns the divisor \text{divi}_1 minus \text{divi}_2.

8.5.5 \( \ast \) (for IsInt, IsToricDivisor)

- \( \ast(k, \text{divi}) \) (operation)
  - \textbf{Returns:} a divisor
  - Returns \( k \) times the divisor \text{divi}.

8.5.6 MonomsOfCoxRingOfDegree (for IsToricVariety, IsHomalgElement)

- MonomsOfCoxRingOfDegree(\text{vari}, \text{elem}) (operation)
  - \textbf{Returns:} a list
  - Returns the monoms of the Cox ring of the variety \text{vari} with degree equal to the class group element \text{elem}. The variable \text{elem} can also be a list.

8.5.7 DivisorOfGivenClass (for IsToricVariety, IsHomalgElement)

- DivisorOfGivenClass(\text{vari}, \text{elem}) (operation)
  - \textbf{Returns:} a divisor
  - Computes a divisor of the variety \text{divi} which is member of the divisor class presented by \text{elem}. The variable \text{elem} can be a homalg element or a list presenting an element.

8.5.8 AddDivisorToItsAmbientVariety (for IsToricDivisor)

- AddDivisorToItsAmbientVariety(\text{divi}) (operation)
  - Adds the divisor \text{divi} to the Weil divisor list of its ambient variety.

8.5.9 Polytope (for IsToricDivisor)

- Polytope(\text{divi}) (operation)
  - \textbf{Returns:} a polytope
  - Returns the polytope of the divisor \text{divi}. Another name for PolytopeOfDivisor for compatibility and shortness.
8.5.10 CoxRingOfTargetOfDivisorMorphism (for IsToricDivisor, IsString)

\[ \text{CoxRingOfTargetOfDivisorMorphism(divi, string)} \]

Returns: a ring

Given a toric divisor \( \text{divi} \), it induces a toric morphism. The target of this morphism is a toric variety. This method returns the Cox ring of this target. The variables are named according to \( \text{string} \).

8.6 Constructors

8.6.1 DivisorOfCharacter (for IsHomalgElement, IsToricVariety)

\[ \text{DivisorOfCharacter(elem, vari)} \]

Returns: a divisor

Returns the divisor of the toric variety \( \text{vari} \) which corresponds to the character \( \text{elem} \).

8.6.2 DivisorOfCharacter (for IsList, IsToricVariety)

\[ \text{DivisorOfCharacter(lis, vari)} \]

Returns: a divisor

Returns the divisor of the toric variety \( \text{vari} \) which corresponds to the character which is created by the list \( \text{lis} \).

8.6.3 CreateDivisor (for IsHomalgElement, IsToricVariety)

\[ \text{CreateDivisor(elem, vari)} \]

Returns: a divisor

Returns the divisor of the toric variety \( \text{vari} \) which corresponds to the Weil group element \( \text{elem} \) by the list \( \text{lis} \).

8.6.4 CreateDivisor (for IsList, IsToricVariety)

\[ \text{CreateDivisor(lis, vari)} \]

Returns: a divisor

Returns the divisor of the toric variety \( \text{vari} \) which corresponds to the Weil group element which is created by the list \( \text{lis} \).
Index

/*
   for IsInt, IsToricDivisor, 32
   for IsToricVariety, IsToricDivisor, 12
*/
+ for IsToricDivisor, IsToricDivisor, 32
- for IsToricDivisor, IsToricDivisor, 32
AddDivisorToItsAmbientVariety
   for IsToricDivisor, 32
AffineCone
   for IsToricVariety, 21
AffineOpenCovering
   for IsToricVariety, 9
AmbientToricVariety
   for IsToricDivisor, 30
   for IsToricSubvariety, 16
AmpleDivisor
   for IsToricVariety and HasPolytopeOfVariety, 22
BasisOfGlobalSections
   for IsToricDivisor, 30
BlowUpOnIthMinimalTorusOrbit
   for IsToricVariety, IsInt, 13
CartierData
   for IsToricDivisor, 29
CartierTorusInvariantDivisorGroup
   for IsToricVariety, 11
CharacterLattice
   for IsToricVariety, 10
CharacterOfPrincipalDivisor
   for IsToricDivisor, 29
CharactersForClosedEmbedding
   for IsToricDivisor, 31
CharacterToRationalFunction
   for IsHomalgElement, IsToricVariety, 13
ClassGroup
   for IsToricMorphism, 25
   for IsToricVariety, 9
ClassOfDivisor
   for IsToricDivisor, 30
ClosureOfTorusOrbitOfCone
   for IsToricVariety, IsCone, 16
Cone
   for IsToricVariety, 18
ConeOfVariety
   for IsToricVariety, 18
CoordinateRing
   for IsAffineToricVariety, 18
   for IsToricVariety, IsList, 18
CoordinateRingOfTorus
   for IsToricVariety, 10
   for IsToricVariety, IsList, 12
CotangentSheaf
   for IsToricVariety, 12
CoxRing
   for IsToricVariety, 9
   for IsToricVariety, IsString, 13
CoxRingOfTargetOfDivisorMorphism
   for IsToricDivisor, 31
   for IsToricDivisor, IsString, 33
CoxVariety
   for IsToricVariety, 11
CreateDivisor
   for IsHomalgElement, IsToricVariety, 33
   for IsList, IsToricVariety, 33
DegreeOfDivisor
   for IsToricDivisor, 31
Dimension
   for IsToricVariety, 10
DimensionOfTorusfactor
   for IsToricVariety, 10
DivisorOfCharacter
   for IsHomalgElement, IsToricVariety, 33
   for IsList, IsToricVariety, 33
DivisorOfGivenClass
34
for IsToricVariety, IsHomalgElement, 32
EulerCharacteristic
   for IsToricVariety, 12
Factors
   for IsToricVariety, 13
Fan
   for IsToricVariety, 13
FanOfVariety
   for IsToricVariety, 11
HasNoTorusfactor
   for IsToricVariety, 8
HasTorusfactor
   for IsToricVariety, 8
InclusionMorphism
   for IsToricSubvariety, 16
IntegerForWhichIsSureVeryAmple
   for IsToricDivisor, 30
IrrelevantIdeal
   for IsToricVariety, 11
IsAffine
   for IsToricVariety, 8
IsAffineToricVariety
   for IsToricVariety, 17
IsAmple
   for IsToricDivisor, 29
IsBasepointFree
   for IsToricDivisor, 29
IsCartier
   for IsToricDivisor, 28
IsCategoryOfToricVarieties
   for IsHomalgCategory, 7
IsClosedSubvariety
   for IsToricSubvariety, 15
IsComplete
   for IsToricVariety, 8
IsDirectProductOfPNs
   for IsToricVariety, 21
IsMorphism
   for IsToricMorphism, 24
IsNormalVariety
   for IsToricVariety, 8
IsNumericallyEffective
   for IsToricDivisor, 29
IsOpen