UNIVERSITÀ DEGLI STUDI DI “ROMA TRE”
DIPARTIMENTO DI MATEMATICA

Smooth complete toric varieties: an algorithmic approach

Anna Scaramuzza

THESIS

Advisors:

Prof. Lucia Caporaso
Dott. Cinzia Casagrande

Ciclo XVIII

Contents

Introduction

The subject of this thesis is to study the properties of smooth, complete toric varieties in a computational way. This means that we build some algorithms whose answers give information about the geometry of the varieties.

A toric variety X of dimension n is a normal algebraic variety that contains a torus $T \cong (\mathbb{C}^*)^n$ as open subset, with an algebraic action of T on X extending the natural multiplication on T.

We always assume X smooth and complete.

Consider $N = \mathrm{Hom}_{\mathbb{Z}}(\mathbb{C}^*, T) \cong \mathbb{Z}^n$ and $N_{\mathbb{Q}} = N \otimes_{\mathbb{Z}} \mathbb{Q}$. Then X is determined by its fan Σ_X in $N_{\mathbb{Q}}$, which is a finite collection of convex rational polyhedral cones in $N_{\mathbb{Q}}$. Geometric properties of X correspond to combinatorial properties of Σ_X. These can be easily translated in equivalent computational ways.

Given X, we deal with the characterization of the group of 1-cycles in X modulo numerical equivalence, denoted $\mathcal{N}_1(X)$ and the associated vector space $\mathcal{N}_1(X)_{\mathbb{Q}} = \mathcal{N}_1(X) \otimes_{\mathbb{Z}} \mathbb{Q}$. Recall that $\mathcal{N}_1(X)_{\mathbb{Q}}$ contains the Mori cone $\mathrm{NE}(X)$, the convex cone generated by classes of effective curves. We are able to compute a basis of $\mathcal{N}_1(X)$, whose elements are classes of invariant curves and to describe $\mathrm{NE}(X)$ giving the coordinates of all classes of invariant curves with respect to the fixed basis.

Through the description of the Mori cone we are able to determine whether X is projective and to find its contractible classes. This means that we can describe all surjective toric morphisms $f : X \to Y$ with connected fibers and with $\rho_X - \rho_Y = 1$, where ρ_X and ρ_Y are the Picard number of X and Y respectively.

For a projective variety X, we also determine the extremal classes among the contractible classes of X. These correspond to the morphisms $f : X \to Y$ as above, with moreover Y projective.

Finally we can compute the invariant blow-up of X along an invariant subvariety V. Conversely, we can establish whether X is the blow-up of another toric variety Y and compute Y. This is important because any X as above can be obtained by blow-ups and blow-downs from $\mathbb{P}^n$ (see [AMR], [Ma], [Mo]).

We choose the programming language *Mathematica 5.0* to write the algorithms.

A smooth complete toric variety X can be described by its fan either giving a complete description of Σ_X or using the language of primitive relations introduced by Batyrev in [Ba2] and [Ba4]. Our algorithms work with both these representations of Σ_X.

We choose either description for X according to the case that we want to study.

Our main tools are the toric Mori theory, the properties of primitive relations, and the theory of resolution of systems of linear Diophantine equations and inequalities.

In order to show how one can use our programs to study a toric variety, we describe some explicit examples and applications in Chapter 7.

Moreover Appendix A is a user's guide to our programs; one just needs to have *Mathematica* and the file `toricvar.m` containing the programs.

We observe that there are some computational problems when working with a big Picard number, in other words, when the number of generators of the fan Σ_X is big. The reason is that we have to solve systems of linear Diophantine equations whose number of variables grows with the number of generators of the fan.

Now, we describe in more detail the algorithms.

DESCRIPTION OF A SMOOTH COMPLETE TORIC VARIETY

Let X be a smooth complete toric variety of dimension n. Its fan Σ_X is completely described by its maximal cones and the set of generators $G(\Sigma_X)$, which are the primitive elements in all 1-dimensional cones in Σ_X. We fix an identification $N \cong \mathbb{Z}^n$, so that the generators are vectors in $\mathbb{Z}^n$.

Primitive collections and primitive relations are important when studying X, in particular in relation with Mori theory. We recall that:

a subset $P \subseteq G(\Sigma_X)$ is called primitive collection if P does not generate a cone in Σ_X, while any proper subset of Pgenerates a cone in Σ_X. For $P = \{x_1, \ldots, x_k\}$ we have a unique relation, called primitive relation associated to P:

$$x_1 + \cdots + x_k - (b_1 y_1 + \cdots + b_h y_h) = 0,$$

where $y_1, \ldots, y_h$ generate a cone in Σ_X and b_i are positive integers for $i = 1, \ldots, h$.

Given Σ_X, it is easy to find its primitive collections; our goal is to compute the relations.

Let's fix a primitive collection $P = \{x_1, \ldots, x_k\}$. We define the linear system associated to P as the system:

$$\sum_{y \in G(\Sigma_X) \setminus P} v_y y \quad = \quad \underline{u}, \tag{1}$$

where $\underline{u} = x_1 + \cdots + x_k$, v_y are the unknowns of the system, y is a generator in $G(\Sigma_X) \setminus P$.

We look for solutions in $\mathbb{Z}^p_{\geq 0}$, where $p = \mathrm{card}(G(\Sigma_X) \setminus P)$.

In [AC] and in [CD] we find the theory to study the systems of linear Diophantine equations and the techniques to find all positive integers which are solutions of a system of linear Diophantine equations. In general, given a $p \times n$ matrix C with entries in $\mathbb{Z}_{\geq 0}$ and a vector $\underline{u}$ in $\mathbb{Z}^n$, then $\underline{v}C = \underline{u}$ is a non-homogeneous system of Diophantine equations with p unknowns represented by the vector $\underline{v} = (\underline{v}_1, \ldots, \underline{v}_p)$. To characterize the set of solution of $\underline{v}C = \underline{u}$, denoted $\mathrm{Sol}(\underline{v}C = \underline{u})$, we fix the following partial order in $\mathbb{Z}^p_{\geq 0}$:

$$(a_1, \ldots, a_p) \leq (b_1, \ldots, b_p) \quad \text{if and only if} \quad a_i \leq b_i \; \forall i \in \{1, \ldots, p\}.$$

Then $\mathrm{Sol}(\underline{v}C = \underline{u})$, is characterized by two finite sets: the set of minimal solutions $\mathrm{Min}(\underline{v}C = \underline{u})$ of $\underline{v}C = \underline{u}$ and the set of minimal non-zero solutions $\mathrm{Min}(\underline{v}C = \underline{0})$ of the associated homogeneous system with respect to the fixed order. In fact, every solution $\underline{a} = (a_1, \ldots, a_p)$ can be written as:

$$\underline{a} \;=\; \underline{\mu} + \sum_{i=1}^{t} \lambda_i \underline{\nu}_i, \qquad (2)$$

where $\underline{\mu}$ is a particular minimal solution in $\mathrm{Min}(\underline{v}C = \underline{u})$, $\{\underline{\nu}_1, \ldots, \underline{\nu}_t\} = \mathrm{Min}(\underline{v}C = \underline{0})$ and $\lambda_i \in \mathbb{Z}_{\geq 0}$.

The set $\mathrm{Min}(\underline{v}C = \underline{u})$ has a central part in finding primitive relations because we show that:

Theorem 2.2.1 Chapter 2. *Let $P = \{x_1, \ldots, x_k\}$ be a primitive collection. The positive integers $b_1, \ldots, b_h$, such that $x_1 + \cdots + x_k - (b_1 y_1 + \cdots + b_h y_h) = 0$ is the primitive relation associated to P, give a minimal solution of the system associated to P.*

Given P, previous theorem allows to restrict our attention to the finite set of minimal solutions of the linear system (1).

Mathematica has a built-in function which solves the system of linear Diophantine equations and gives all possible solutions as in (2), (see [Wo]). We build a procedure giving the set of minimal solutions. In order to find which is the minimal solution that gives the primitive relation we use the following:

Proposition 2.2.3 Chapter 2. *In the set of minimal solutions of the system associated to P there is a unique minimal solution $\underline{m} = (m_y)_{y \in G(\Sigma_X) \setminus P} \in \mathbb{Z}^p_{\geq 0}$ such that the set $G(\underline{m}) = \{y \in G(\Sigma_X) \setminus P \mid m_y \neq 0\}$ generates a cone in Σ_X.*

A solution as in the previous proposition is called optimal solution and it determines the primitive relation associated to P.

Using Theorem 2.2.1 and Proposition 2.2.3 in Chapter 2, we build the algorithm `PrimRel` which computes, for every primitive collection P, its primitive relation.

When X is projective, the set of primitive relations determines X up to isomorphism. Let's see how we can compute Σ_X explicitly. First, the maximal cones are the subsets of cardinality n which do not contain a primitive collection. Then, we fix a basis B of N (given by the generators of a maximal cone) and interpret the list of all primitive relations as a linear system of equations over $\mathbb{Z}^{\mathrm{card}(G(\Sigma_X))}$ whose unknowns are the coordinates of the generators in $G(\Sigma_X) \setminus B$. Solving the system, we obtain the coordinates of all generators of $G(\Sigma_X)$ with respect to B. Computationally, this process is simpler than the other because it solves just one system of linear equations.

TORIC BLOW-UPS AND BLOW-DOWNS

The technique described above to determine the primitive relations is very useful also to describe the blow-up Y of X along an invariant subvariety $V = V(\sigma)$ of X:

$$\psi : Y \to X$$

The fan of Y is easily computed from the fan of X. We would like to be able to compute the primitive relations of Y from those of X.

The main reference about the argument is Theorem 4.3 in [Sa], which says how the primitive collections of Y can be computed by primitive collections of X.

There are four types of primitive collections for Y. For three of them, the primitive relations can be determined rather easily from ones of X. For the last group of primitive collections, we have to use the algorithm described in the previous paragraph. This is computationally longer, because it involves solving systems of linear Diophantine equations.

The situation is simpler if we know Y and want to determine X. In this case we can compute all primitive relations of X starting from those of Y and we do not need to solve systems of linear Diophantine equations.

PROJECTIVITY AND MORI CONE

The description of Σ_X with primitive relations is useful to characterize the projectivity of X. Let's consider the Mori cone and denote by $\mathrm{NE}(X)_{\mathbb{Z}}$ the intersection $\mathrm{NE}(X) \cap \mathbb{Z}$.

Reid studies the toric Mori theory in [R]. He proves that $\mathrm{NE}(X)_{\mathbb{Z}}$ is generated as a semigroup by the set $\mathcal{I}$ of all classes of invariant curves, hence $\mathcal{I}$ generates $\mathcal{N}_1(X)$ as a group. Then the set $\mathcal{I}$ contains a basis B of $\mathcal{N}_1(X)$.

$\mathcal{I}$ is a set of 1-cycles and we interpret it as a set of linear polynomials in the generators of Σ_X. If $G(\Sigma_X) = \{x_1, \ldots, x_t\}$, then $\mathcal{I} \subset \mathbb{Z}[x_1, \ldots, x_t]_1$. In *Mathematica*, the built-in function `Eliminate` computes a system of generators of the set of syzygies $\mathrm{Syz}(\mathcal{I})$ among the elements in $\mathcal{I}$. We have written an algorithm to compute a basis $B \subset \mathcal{I}$ of $\mathcal{N}_1(X)$ using the generators of $\mathrm{Syz}(\mathcal{I})$.

Replacing in the syzygies the elements of B with the vectors e_i of the canonical basis of $\mathbb{Q}^{\rho_X}$, we obtain the vector $\underline{w}_\gamma$ of the coordinates of every class $\gamma \in \mathcal{I}$ with respect to B. In this way, the Mori cone can be seen in $\mathbb{Q}^{\rho_X}$ as the cone generated by the vectors $\underline{w}_\gamma$, for $\gamma \in \mathcal{I}$.

Under this identification, we want to use the characterization of projectivity given by Kleiman's criterion of ampleness (see [Kl]): *X is projective if and only if $\mathrm{NE}(X)$ is strongly convex.*

Then the condition of projectivity is translated in the following computational way:

Lemma 4.3.1 Chapter 4. *X is projective if and only if the equation*

$$\sum_{i=1}^{d} v_i \underline{w}_i = 0$$

has only the trivial solution in $\mathbb{Z}^d_{\geq 0}$.

Finally we are able to compute the set of contractible classes of X. Let $\gamma \in \mathcal{I}$. It is *contractible* if there exist a toric variety X_γ and an equivariant morphism $\varphi_\gamma : X \to X_\gamma$, surjective and with connected fibers, such that for every irreducible curve $C \subset X$,

$$\varphi_\gamma(C) = \{pt\} \iff [C] \in \mathbb{Q}_{\geq 0}\gamma.$$

We denote by $\mathcal{C}$ the set of all contractible classes. We can easily determine $\mathcal{C}$ using the criterion 4.1.3 (see also [C2], [Sa]), in terms of the primitive relations of X.

The Mori Cone in the projective case

When X is projective, we know that the set $\mathcal{C}$ of contractible classes generates $\mathrm{NE}(X)_{\mathbb{Z}}$ as a semigroup, hence it contains a basis B of $\mathcal{N}_1(X)$.

With the same techniques described in the previous paragraph, we can find such a basis B and compute the coordinates of all elements of $\mathcal{C}$ with respect to B.

In this case we are interested in finding the extremal classes of $\mathrm{NE}(X)$, that is, the primitive elements in $R \cap \mathrm{NE}(X)_{\mathbb{Z}}$, where R is a 1-dimensional face of $\mathrm{NE}(X)$.

We know by Reid's results that every extremal class is contractible. We remark that the set $\mathcal{E}$ of all extremal classes generates $\mathrm{NE}(X)$ over $\mathbb{Q}$, however it is not known whether it generates $\mathrm{NE}(X)_{\mathbb{Z}}$ as a semigroup.

In order to determine which contractible classes are extremal, we use the following:

Proposition 4.4.1 Chapter 4. *Let $\mathcal{C} = \{\gamma_1, \dots, \gamma_s\}$ be the set of all contractible classes. Let $\{\underline{w}'_1, \dots, \underline{w}'_s\}$ be the set of vectors of the coordinates of all contractible classes with respect to a basis B in $\mathcal{C}$. Fixed $i \in \{1, \dots, s\}$, then γ_i is non extremal if and only if the equation*

$$\underline{w}'_i = \sum_{j \neq i} v_j \underline{w}'_j$$

has a non-trivial solution in $\mathbb{Z}^{s-1}_{\geq 0}$.

Then we can compute $\mathcal{E}$ by solving the linear system above.

Arguments of Chapters

In **Chapter 1** we find an introduction of concepts which we use to study smooth, complete, toric varieties. More precisely, we give the definitions of fan of X, of primitive relations, of the group $\mathcal{N}_1(X)$ and of the Mori cone of a smooth complete toric variety X. Moreover we give the main results about these arguments. Then, we consider the case of a smooth projective toric variety introducing the contractible classes of curves.

The theory of systems of linear Diophantine equations is explained in **Chapter 2**. Here we distinguish the case of homogeneous and non-homogeneous systems and in both cases we analyze the sets of solutions characterizing them with the sets of minimal solutions. Then, we explain how this argument is related with the toric varieties and more explicitly with the primitive relations of a toric variety X. This theory is used to build the algorithm `PrimRel`. The algorithm and the commands that we use to write it are described in the last sections of this chapter.

In **Chapter 3**, we introduce the ways of representing a smooth, complete toric variety X and the algorithms which describe X in *Mathematica*.

Chapter 4 is devoted to the classes of curves and their representation in the vector space $\mathcal{N}_1(X)_{\mathbb{Q}}$ of a smooth complete toric variety X. Here we explain how to associate a 1-cycle in $\mathcal{N}_1(X)$ and how to find the contractible classes of curves of X. In Section 2 we describe the techniques used to find a basis B of $\mathcal{N}_1(X)$ as a group and the algorithm which computes B and the coordinates of every class of curves with respect to B. In Section 3 we consider the problem of the projectivity of X and the computational formulation of it. Then, we describe the algorithm and the commands used to write the program `ProjQ`. Finally, we consider a smooth projective toric variety X and explain how we can find the extremal classes of curves among the contractible classes of curves.

In **Chapter 5** we present the theory describing the algorithm which computes the blow-up Y of X along an invariant subvariety V. The statements touch the problem of describing Y with its primitive relations and of computing them from primitive relations of X. In this case we have to use the techniques explained in Chapter 2 to compute some primitive relations.

Chapter 6 is devoted to the description of algorithm `Blowdown` and to the statements used to write it. In this case we do not need to use the systems of linear Diophantine equations to describe the variety obtained by the blow-down with primitive relations.

In **Chapter 7**, we present some interesting examples and applications. We describe them with our programs. In this way we test the algorithms and see that in most cases they work very quickly.

In **Appendix A**, we describe all algorithms of the package "*Toric Varieties*" and we explain how one can use them. For every command, we specify its input and output, we say what it does and give an example.

Chapter 1

Basic Notions

A toric variety X of dimension n is a normal complex algebraic variety that contains an algebraic group T isomorphic to $(\mathbb{C}^*)^n$ (a torus), as a dense open subset, with an algebraic action $T \times X \to X$ of T on X which extends the natural action of T on itself (multiplication in T).

This definition is a theoretic and does not allow to have a "computational" description of a toric variety, so we introduce the concept of fan to give an equivalent, but combinatorial definition of toric variety.

A fan can be represented by either the list of its maximal cones and generators or by the list of primitive relations. Its combinatorial properties correspond to geometric properties of X.

Moreover we are interested in knowing if a smooth complete toric variety X is projective, so we introduce the toric Mori theory.

The chapter is divided in five sections: in Section 1.1 we give the definitions of fan and cone and some their properties. In Section 1.2 we analyze the correspondence among cones and toric subvarieties of X, in Section 1.3 we introduce the concepts of primitive collections and primitive relations. A more detailed introduction about these arguments can be found in [Ba2], [Ba4], [F], [E2] and [O2]. In Section 1.4 we find the main statements about the toric Mori theory for a smooth, complete, toric variety and then, in Section 1.5, we consider the toric Mori theory for smooth, projective, toric varieties. The main references about these arguments are [O2], [R] and [Wi].

1.1 Fans and Cones

Let $N = \mathrm{Hom}_{\mathbb{Z}}(\mathbb{C}^*, T)$ be a free Abelian group of finite rank n, i.e. $N \cong \mathbb{Z}^n$. Given N, we define its dual lattice $M = \mathrm{Hom}_{\mathbb{Z}}(N, \mathbb{Z})$. We also consider the vector space $N_{\mathbb{Q}} = N \otimes_{\mathbb{Z}} \mathbb{Q}$, whose dual space can be identified with $M_{\mathbb{Q}} = M \otimes_{\mathbb{Z}} \mathbb{Q}$. We denote by $(\ , \ )$ the natural pairing on $M_{\mathbb{Q}} \times N_{\mathbb{Q}}$.

Definition 1.1.1. *A* ***rational convex polyhedral cone*** *(or a cone) in $N_{\mathbb{Q}}$ is a set of the form:*

$$\langle x_1, \ldots, x_k \rangle \quad = \quad \left\{ \sum_{i=1}^{k} \lambda_i x_i \mid \lambda_i \geq 0,\ \lambda_i \in \mathbb{Q} \right\},$$

where $\{x_1, \ldots, x_k\} \subset N$ *is a finite set.*

Given a cone σ, we denote by $G(\sigma)$ the unique set $\{y_1, \ldots, y_m\}$ such that:

- $\sigma = \langle y_1, \ldots, y_m \rangle$;
- every y_i is primitive in the lattice N;
- for every $i = 1, \ldots, m$ the cone $\langle y_1, \ldots, \hat{y}_i, \ldots, y_m \rangle$ is strictly smaller than σ.

We call the elements of $G(\sigma)$ the **generators** of σ.

We define dimension of the cone σ as the dimension of the smallest linear subspace $\mathrm{Span}(\sigma)$ containing σ. Moreover we call $\mathrm{RelInt}(\sigma)$ the interior of σ in $\mathrm{Span}(\sigma)$.

Let $\sigma \in N_{\mathbb{Q}}$ be a cone. Its **dual cone** is the set:

$$\sigma^{\vee} \quad = \quad \{y \in M_{\mathbb{Q}} \mid (y, x) \geq 0 \text{ for all } x \in \sigma \}.$$

Proposition 1.1.2 (Farkas' Theorem, [F], page 11)**.** *The dual cone of a convex polyhedral cone is again a convex polyhedral cone.*

A cone σ is **strongly convex** if $\sigma \cap (-\sigma) = \{0\}$. There are equivalent characterizations of strong convexity.

Proposition 1.1.3 ([F], page 14)**.** *Let* $\sigma \subset N_{\mathbb{Q}}$ *be a rational convex polyhedral cone. Then the following are equivalent:*

1. σ *is strongly convex;*
2. σ *contains no positive-dimensional linear subspace;*
3. $\dim(\sigma^{\vee})=n$.

Definition 1.1.4. *A convex rational polyhedral cone* $\sigma \subset N_{\mathbb{Q}}$ *is **regular** if its generators form a part of a* $\mathbb{Z}$*-basis of* N.

Toric varieties are described by a set of strongly convex rational polyhedral cones called fan.

Definition 1.1.5. *A **fan*** Σ *in* $N_{\mathbb{Q}}$ *is a finite collection of strongly convex rational polyhedral cones such that:*

1. *each face of a cone in* Σ *belongs to* Σ*;*
2. *the intersection of two cones in* Σ *is a face of each.*

For every $i \in \{0, \ldots, n\}$, $\Sigma(i)$ denotes the set of cones in Σ of dimension i.

For every cone $\sigma \in \Sigma(1)$, let $x_\sigma \in \sigma \cap N$ be its primitive generator. The set $G(\Sigma) = \{x_\sigma \mid \sigma \in \Sigma(1)\}$ is the set of **generators** of the fan.

Clearly, a fan Σ is determined by $G(\Sigma)$ and by the set of its maximal cones.

Given a fan Σ, let X be the associated toric variety. We will then write $\Sigma = \Sigma_X$. Then geometric properties of X correspond to combinatorial properties of Σ_X.

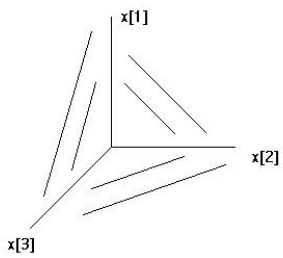
A toric variety is **smooth** if and only if every cone $\sigma \subset \Sigma$ is regular. We also say that the fan is regular.

In this thesis we will deal with smooth and complete toric varieties. The completeness of a toric variety is another geometric property that we can read on the fan: X is **complete** if and only if

$$\bigcup_{\sigma \in \Sigma_X} \sigma = N_{\mathbb{Q}}.$$

Example 1.1.6. Consider $N = \mathbb{Z}^2$ and consider the points $x_1 = (1,0)$, $x_2 = (0,1)$, $x_3 = (-1,-1)$ in $N_{\mathbb{Q}}$. Let Σ be the fan with $G(\Sigma) = \{x_1, x_2, x_3\}$ and $\Sigma(2) = \{\langle x_1, x_2\rangle, \langle x_1, x_3\rangle, \langle x_2, x_3\rangle\}$.

All maximal cones are regular and the union is $N_{\mathbb{Q}}$. The variety X is $\mathbb{P}^2$.



The fan Σ_X determines X up to isomorphism in the following sense:

Theorem 1.1.7. *Let X and Y be two smooth and complete toric varieties with fans Σ_X and Σ_Y in the vector spaces $(N_X)_{\mathbb{Q}}$ and $(N_Y)_{\mathbb{Q}}$ respectively.*

Then X and Y are isomorphic as abstract varieties if and only if there exists a lattice isomorphism $\varphi : N_X \to N_Y$ such that for every $\sigma \in \Sigma_X$, $\varphi(\sigma) \in \Sigma_Y$.

Proof. It is shown in [Ba4], Theorem 2.2.4 that the existence of an abstract isomorphism implies the existence of an equivariant isomorphism (notice that the Fano assumption plays no role in the proof). The rest is standard toric geometry (see [O2], Section 1.5). □

1.2 Correspondence among cones in Σ_X and invariant subvarieties of X

Let X be a toric variety with fan Σ_X.

There is a bijection between Σ_X and the set of orbits of the torus T in X. If $\sigma \in \Sigma_X$, we denote by $O(\sigma)$ the corresponding orbit in X, then $O(\sigma) \cong (\mathbb{C}^*)^{n-\dim(\sigma)}$.

We will denote by $V(\sigma)$ the Zariski closure of the orbit $O(\sigma)$ in X. In the case of a one dimensional cone $\langle x \rangle$, we use the notation $V(x)$ for the subvariety. The subvarieties $V(\sigma)$ are closed with respect to the torus action and we will refer to them as invariant subvarieties.

The following proposition gives a more concrete way to describe this correspondence:

Proposition 1.2.1 ([O2], Corollary 1.7 on page 11)**.** *Let N_σ be the sub-lattice of N generated by $\sigma \cap N$ for every $\sigma \in \Sigma$ and let $N(\sigma) = N/N_\sigma$ be the quotient. For every $\tau \in \Sigma$ such that $\sigma \subsetneq \tau$, let $\bar{\tau}$ be the image of σ in the quotient vector space $N(\sigma)_{\mathbb{Q}} = N_{\mathbb{Q}}/\mathrm{Span}(\sigma)$. Then*

$$\Sigma(\sigma) = \{\overline{\tau} \mid \tau \in \Sigma,\ \sigma \subset \tau\ \}$$

is a fan in $N(\sigma)_{\mathbb{Q}}$ and $X(\Sigma(\sigma))$ is isomorphic to $V(\sigma)$.

Example 1.2.2. Fix $N = \mathbb{Z}^3$ and in $N_{\mathbb{Q}}$ consider the fan Σ defined by following generators

$$\begin{array}{llll} x_1 = (1,0,0), & x_2 = (0,1,0), & x_3 = (0,0,1), & x_4 = (-1,-1,-1), \\ x_5 = (-1,0,-1), & x_6 = (-1,-1,0), & x_7 = (0,-1,-1), & x_8 = (0,0,-1), \end{array}$$

and maximal cones:

$$\begin{array}{llll} \langle x_1, x_2, x_3 \rangle, & \langle x_1, x_2, x_8 \rangle, & \langle x_1, x_3, x_6 \rangle, & \langle x_1, x_6, x_7 \rangle, \\ \langle x_1, x_7, x_8 \rangle, & \langle x_2, x_3, x_5 \rangle, & \langle x_2, x_5, x_8 \rangle, & \langle x_3, x_5, x_6 \rangle, \\ \langle x_4, x_5, x_6 \rangle, & \langle x_4, x_5, x_7 \rangle, & \langle x_4, x_6, x_7 \rangle, & \langle x_5, x_7, x_8 \rangle. \end{array}$$

The variety X associated to Σ is the blow-up of $\mathbb{P}^3$ along four invariant curves.

We consider the cone $\langle x_8 \rangle$ and the corresponding subvariety Z. The maximal cones containing x_8 are:

$$\langle x_1, x_2, x_8 \rangle, \quad \langle x_1, x_7, x_8 \rangle, \quad \langle x_2, x_5, x_8 \rangle, \quad \langle x_5, x_7, x_8 \rangle.$$

Let π be the projection $\pi : N \to N(\langle x_8 \rangle)$ and let $\overline{x}_i$ be the image of every generator x_i under π. Then, in $N(\langle x_8 \rangle) \otimes \mathbb{Q}$ the fan of Z is:

$$\langle \overline{x}_1, \overline{x}_2 \rangle, \quad \langle \overline{x}_1, \overline{x}_7 \rangle, \quad \langle \overline{x}_2, \overline{x}_5 \rangle, \quad \langle \overline{x}_5, \overline{x}_7 \rangle,$$

and generators are:

$$\overline{x}_1 = (1,0), \quad \overline{x}_2 = (0,1), \quad \overline{x}_5 = (-1,0), \quad \overline{x}_7 = (0,-1).$$

Z is isomorphic to $\mathbb{P}^1 \times \mathbb{P}^1$.

1.3 Primitive collections and primitive relations

Let X be a smooth, complete, toric variety of dimension n. Let Σ_X be the fan associated to X.

We recall the definitions of primitive collections and relations introduced by Batyrev in [Ba2] and [Ba4]. They describe toric varieties from another combinatorial point of view.

Definition 1.3.1. *A subset $P \subseteq G(\Sigma_X)$ is a* ***primitive collection*** *if:*

1. *$\langle P \rangle \notin \Sigma_X$;*
2. *for every $x \in P$, the set $P \setminus \{x\}$ generates a cone in Σ_X.*

$PC(\Sigma_X)$ or $PC(X)$ will denote the set of all primitive collections of Σ_X.

Definition 1.3.2. *Let $P = \{x_1, \ldots, x_k\} \subseteq G(\Sigma_X)$ be a primitive collection. Since X is complete, there exists a unique cone $\sigma_P = \langle y_1, \ldots, y_r \rangle$ in Σ_X such that $x_1 + \cdots + x_k \in \mathrm{RelInt}(\sigma_P)$. Hence, there exist unique numbers $a_1, \ldots, a_r \in \mathbb{Z}_{>0}$ such that*

$$x_1 + \cdots + x_k - (a_1 y_1 + \cdots + a_r y_r) \quad = \quad 0. \tag{1.1}$$

This is called ***primitive relation*** *associated to the primitive collection P.*
The number $k - a_1 - \cdots - a_r$ is the ***degree*** *of the primitive relation.*

We will call $x_1 + \cdots + x_k$ and $a_1 y_1 + \cdots + a_r y_r$ respectively **positive part** and **negative part** of the primitive relation.

The cone σ_P is the **cone associated** to the primitive collection P.

Remark 1.3.3. A primitive collection is always a subset of $G(\Sigma_X)$ of cardinality k with $2 \leq k \leq n+1$. In fact the set $\{x_i\} \subset G(\Sigma_X)$ always generates a 1-dimensional cone in Σ_X.

Remark 1.3.4. By definition, it follows that for any subset S of $G(\Sigma_X)$, either S generates a cone in Σ_X, or S contains a primitive collection.

Proposition 1.3.5 ([Ba4], Proposition 3.1)**.** *Let $P \in PC(X)$ be a primitive collection in $G(\Sigma_X)$ and let $G(\sigma_P)$ be the set of generators of the cone associated to P. Then $P \cap G(\sigma_P) = \emptyset$.*

1.4 The Mori cone of a smooth and complete toric variety

Let X be a smooth and complete toric variety and let Σ_X be its fan.

Let $\mathcal{N}_1(X)$ be the group of algebraic 1-cycles on X modulo numerical equivalence and define the vector space $\mathcal{N}_1(X)_{\mathbb{Q}} = \mathcal{N}_1(X) \otimes_{\mathbb{Z}} \mathbb{Q}$.

The Mori cone is the convex cone generated by the classes of effective curves modulo numerical equivalence in $\mathcal{N}_1(X)_{\mathbb{Q}}$:

$$\mathrm{NE}(X) = \left\{ \gamma \in \mathcal{N}_1(X)_{\mathbb{Q}} \;\middle|\; \gamma = \left[\sum a_i C_i\right], \text{ with } a_i \in \mathbb{Q}_{\geq 0} \right\} \subseteq \mathcal{N}_1(X)_{\mathbb{Q}}.$$

We use the notation $\mathrm{NE}(X)_{\mathbb{Z}}$ to denote the intersection of $\mathrm{NE}(X)$ with $\mathcal{N}_1(X)$.

There is an exact sequence:

$$0 \longrightarrow \mathcal{N}_1(X) \xrightarrow{\phi} \mathbb{Z}^t \xrightarrow{\psi} N \longrightarrow 0$$

where $t = \mathrm{card}(G(\Sigma_X))$ and the maps ϕ, ψ are respectively defined by $\gamma \mapsto (\gamma \cdot V(x))_{x \in G(\Sigma_X)}$ and $(a_x)_{x \in G(\Sigma_X)} \mapsto \sum_{x \in G(\Sigma_X)} a_x x$.

Hence, ϕ defines a canonical isomorphism:

$$\mathcal{N}_1(X) \cong \left\{ \sum_{x \in G(\Sigma_X)} a_x x = 0 \;\middle|\; a_x \in \mathbb{Z} \right\}$$

that is $\mathcal{N}_1(X)$ is identified with the lattice of integral relations among elements of $G(\Sigma_X)$.

We will then identify every $\gamma \in \mathcal{N}_1(X)$ with the corresponding relation

$$\sum_{x \in G(\Sigma_X)} (\gamma \cdot V(x))x = 0. \tag{1.2}$$

Let's recall how one can write **the relation associated to the numerical class of an invariant curve**.

Let $\sigma \in \Sigma_X(n-1)$, then $V(\sigma) \subset X$ is an invariant curve. It is isomorphic to $\mathbb{P}^1$. We denote by $\gamma(\sigma)$ the class modulo numerical equivalence of $V(\sigma)$.

Write $\sigma = \langle x_1, \ldots, x_{n-1} \rangle$. Let σ' and σ'' be the two maximal cones in Σ_X such that $\sigma' \cap \sigma'' = \sigma$:

$$\begin{aligned} \sigma' &= \langle x_1, \ldots, x_{n-1}, x_n \rangle, \\ \sigma'' &= \langle x_1, \ldots, x_{n-1}, x_{n+1} \rangle. \end{aligned}$$

Since X is smooth, $x_1, \ldots, x_{n-1}, x_n$ and $x_1, \ldots, x_{n-1}, x_{n+1}$ are two bases of N, hence, there exist uniquely determined integers $a_1, \ldots, a_n$ such that:

$$x_n + x_{n+1} + a_1 x_1 + \cdots + a_{n-1} x_{n-1} = 0. \tag{1.3}$$

By [R], (1.3) is the relation corresponding to $\gamma(\sigma)$.

We can also read the normal bundle of $V(\sigma)$ in (1.3), in fact:

$$\mathcal{N}_{V(\sigma)/X} \cong \bigoplus_{i=1}^{n-1} \mathcal{O}_{\mathbb{P}^1}(a_i).$$

The numerical classes of invariant curves allow to Reid to characterize $\mathrm{NE}(X)_{\mathbb{Z}}$. In fact, he proves:

Theorem 1.4.1 ([R], Corollary 1.7 - [O2] on page 106)**.** *Let X be a smooth complete toric variety of dimension n with fan Σ_X. Then*

$$\mathrm{NE}(X)_{\mathbb{Z}} = \sum_{\sigma \in \Sigma_X(n-1)} \mathbb{Z}_{\geq 0}\gamma(\sigma).$$

This means that the classes $\gamma(\sigma)$ generate $\mathrm{NE}(X)_{\mathbb{Z}}$ as a semigroup. In particular, this says that $\mathrm{NE}(X)$ **is a closed rational polyhedral cone in** $\mathcal{N}_1(X)_{\mathbb{Q}}$.

Observe that by definition every element of $\mathcal{N}_1(X)_{\mathbb{Q}}$ is a linear combination of elements in $\mathrm{NE}(X)$, namely $\dim \mathrm{NE}(X) = \rho_X$.

Proposition 1.4.2 ([C2], Lemma 1.4 - [Kr], Proposition 2.1)**.** *Let $\gamma \in \mathcal{N}_1(X)$ given by the relation*

$$a_1x_1 + \cdots + a_kx_k - (b_1y_1 + \cdots + b_ry_r) = 0,$$

with $a_i, b_j \in \mathbb{Z}_{>0}$ for each i, j. If $\langle y_1, \ldots, y_r\rangle \in \Sigma_X$, then $\gamma \in \mathrm{NE}(X)$.

The primitive relations are relations among the generators of Σ_X, hence they can be interpreted as elements of $\mathcal{N}_1(X)$. Moreover Proposition 1.4.2 says that **for every primitive collection** $P \in PC(X)$ **the primitive relation** $r(P)$ **belongs to** $\mathrm{NE}(X)_{\mathbb{Z}}$.

There is another important type of classes in $\mathrm{NE}(X)$:

Definition 1.4.3. *Let $\gamma \in \mathrm{NE}(X)_{\mathbb{Z}}$ be primitive in $\mathbb{Z}_{\geq 0}\gamma$ and such that there exists some irreducible curve in X having numerical class in $\mathbb{Z}_{\geq 0}\gamma$. We say that γ is* ***contractible*** *if there exist a toric variety X_γ and an equivariant morphism $\varphi_\gamma : X \to X_\gamma$, surjective and with connected fibers, such that for every irreducible curve $C \subset X$,*

$$\varphi_\gamma(C) = \{pt\} \iff [C] \in \mathbb{Q}_{\geq 0}\gamma.$$

Hence contractible classes correspond to "elementary" toric morphisms with connected fibers with target X.

Theorem 2.2 in [C2] says that every contractible class is also a primitive relation and it is always the class of some invariant curve.

Hence, we have three important subsets of $\mathrm{NE}(X)_{\mathbb{Z}}$, which are all finite:

$$\begin{array}{rcl} \mathcal{I} & = & \{\text{classes of invariant curves}\} \\ & & \cup \\ \mathcal{C} & = & \{\text{contractible classes}\} \\ & & \cap \\ \mathrm{PR} & = & \{\text{primitive relations}\} \end{array} \tag{1.4}$$

Let γ be a contractible class and suppose that

$$x_1 + \cdots + x_k - (a_1y_1 + \cdots + a_ry_r) \quad = \quad 0$$

is the primitive relation associated to γ.

We observe that if $r = 0$, that is the relation has the form:

$$x_1 + \cdots + x_k = 0,$$

then X_γ is smooth and $\varphi_\gamma : X \to X_\gamma$ is a $\mathbb{P}^{k-1}$-bundle.

If $r > 0$, then φ_γ is birational and its exceptional locus is $V(\langle y_1, \ldots, y_r \rangle)$. Hence, φ_γ is divisorial if and only if $r = 1$. Moreover, if $r = 1$ and $a_1 = 1$, i. e. if the relation has the form

$$x_1 + \cdots + x_k - y_1 = 0,$$

then X_γ is smooth and φ_γ is the blow-up of an invariant subvariety of codimension k in X_γ.

Every blow-up of a smooth and complete toric variety along an invariant subvariety has this form. In fact, given a toric variety Y and a subvariety $V = V(\langle x_1, \ldots, x_k \rangle)$ in Y, we consider the variety X defined by the blow-up of Y along V. The fan Σ_X of X is obtained by a star subdivision of the cone $\langle x_1, \ldots, x_k \rangle$ and of all cones which contain the cone $\langle x_1, \ldots, x_k \rangle$ (see [O2], Proposition 1.26 on page 38). Then the set of generators of Σ_X is given by the set of generators of fan of Y together with a new generator y. This last is determined by the relation

$$x_1 + \cdots + x_k - y = 0.$$

This is a primitive relation for Σ_X and it is the relation associated to the class of invariant curves contracted by the blow-up.

1.5 The Mori cone in the projective case

Let X be a smooth complete toric variety.

We define the group of divisors in X modulo numerical equivalence: $\mathcal{N}^1(X)$ and the vector space $\mathcal{N}^1(X)_\mathbb{Q} = \mathcal{N}^1(X) \otimes \mathbb{Q}$ obtained from $\mathcal{N}^1(X)$. Inside $\mathcal{N}^1(X)_\mathbb{Q}$, we consider the cone of nef $\mathbb{Q}$-divisors. It is denoted by *Nef*(X).

We observe that $\mathcal{N}^1(X)_\mathbb{Q}$ is dual to $\mathcal{N}_1(X)_\mathbb{Q}$ and that *Nef*(X) is the dual cone of $\mathrm{NE}(X)$. By Proposition 1.1.2, *Nef*(X) **is a polyhedral rational cone**.

By Kleiman's criterion of ampleness (see [Kl]), we know that a divisor D is ample if and only if its numerical class lies in the interior of *Nef*(X).

Then we have:

$$\begin{aligned} X \textit{ is projective} &\iff \textit{there exists an ample divisor } D \\ &\iff \textit{Nef}(X) \textit{ has non-empty interior} \\ &\iff \dim \textit{Nef}(X) = \dim \mathcal{N}^1(X)_\mathbb{Q} = \rho_X. \end{aligned}$$

By Proposition 1.1.3 we get:

$$X \textit{ is projective} \iff \mathrm{NE}(X) \textit{ is strongly convex.}$$

Assume now that X is projective. We call **extremal ray** a 1-dimensional face R of $\mathrm{NE}(X)$ and **extremal class** the primitive element of $R \cap \mathrm{NE}(X)_{\mathbb{Z}}$. Let $\mathcal{E}$ be the set of all extremal classes of X.

Then we can reformulate Reid's results in toric Mori theory as follows:

Theorem 1.5.1 ([R], Theorem 1.5)**.** *Let X be a projective, smooth, toric variety. Any extremal class is contractible.*

The difference among contractible and extremal classes is given by the proposition:

Proposition 1.5.2 ([Bo], Lemma 1 - [C2], Corollary 3.3)**.** *Let X be a projective, smooth, toric variety. Let $\gamma \in \mathrm{NE}(X)$ be a contractible class and let $\varphi_\gamma : X \to X_\gamma$ be the associated morphism. Then γ is not extremal if and only if φ_γ is birational and the variety X_γ is not projective.*

Finally, we notice that by definition of extremal classes, we have

$$\mathrm{NE}(X) = \sum_{\gamma \in \mathcal{E}} \mathbb{Q}_{\geq 0} \gamma$$

but it is not known whether the same holds over $\mathbb{Z}$, i. e. whether the set $\mathcal{E}$ generates $\mathrm{NE}(X)_{\mathbb{Z}}$ as semigroup. If we consider all contractible classes we have:

Theorem 1.5.3 ([C2])**.** *Let X be a projective, smooth, toric variety. Let $\mathcal{C}$ be the set of all contractible classes. Then*

$$\mathrm{NE}(X)_{\mathbb{Z}} = \sum_{\gamma \in \mathcal{C}} \mathbb{Z}_{\geq 0} \gamma$$

that is $\mathcal{C}$ generates $\mathrm{NE}(X)_{\mathbb{Z}}$ as a semigroup.

Chapter 2

Primitive relations and systems of linear Diophantine equations

In this chapter we present some known results about systems of linear Diophantine equations. In fact, given a smooth and complete toric variety X, described by the coordinates of generators and by maximal cones, we can get its primitive relations solving a system of linear Diophantine equations.

In [AC] there is a detailed description of the new algorithm for solving linear Diophantine systems of both equations and inequations. There, the authors remark that their algorithm solves a linear Diophantine system of both equations and inequations directly, that is, without adding slack variables for encoding inequalities as equations. Hence, it is a generalization of the algorithm due to Contejean and Devie for solving systems of linear Diophantine equations (see [CD]).

The chapter is divided in four sections. In Section 2.1 we introduce the systems of linear Diophantine equations analyzing the cases of homogeneous and non-homogeneous systems. In both cases the set of solutions can be described only with the finite sets of minimal solutions with respect to fixed partial order. Section 2.2 is devoted to relate the primitive relations with systems of linear Diophantine equations. More precisely, we prove that the coefficients of the negative part of every primitive relation $r(P)$ are a minimal solution of the system associated to P. In the last two sections we present the commands and the algorithm which compute a primitive relation of a primitive collection P.

The main references about systems of linear Diophantine equations are [AC], [CD]. In [CP] we find all references about the theory of semigroups used to characterize the set of solutions of a system of linear Diophantine equations. In [dps] and [Wo] we find a detailed introduction to the commands and to the techniques used in *Mathematica* to solve the systems of linear Diophantine equations. For other related algebraic arguments we refer to [CLO1], [CLO2], [L] and [Se].

2.1 Systems of linear Diophantine equations

Let's consider a system of linear Diophantine equations:

$$\underline{v}C \quad = \quad \underline{u} \tag{2.1}$$

where $\underline{v} = (v_1, \ldots, v_p)$ is the vector of unknowns, $C = (c_{ij})$ is a $p \times n$ matrix with $c_{ij} \in \mathbb{Z}$, and $\underline{u}$ is a vector in $\mathbb{Z}^n$.

We are going to search all solutions whose coordinates are non-negative integers. Hence, the solutions of the system are in $\mathbb{Z}^p_{\geq 0}$.

We will distinguish whether the linear system is homogeneous or not.

Observe that when the system is homogeneous, the set of all solutions in $\mathbb{Z}^p_{\geq 0}$ has the structure of a semigroup.

Ajili and Contejean explain (see [AC]) that using the theory of semigroups we can characterize the set of all solutions of a system with a finite set of solutions.

Consider the semigroup of natural numbers $(\mathbb{Z}^p_{\geq 0}, +)$ and fix in $\mathbb{Z}^p_{\geq 0}$ the following order:

$$(a_1, \ldots, a_p) \leq (b_1, \ldots, b_p) \quad \text{if and only if } a_i \leq b_i \quad \forall\, i \in \{1, \ldots, p\}.$$

Remark 2.1.1. This is a partial order on $(\mathbb{Z}^p_{\geq 0}, +)$ and the element $\underline{0} \in \mathbb{Z}^p_{\geq 0}$ is the minimum with respect to this order. So given a decreasing chain of elements in $\mathbb{Z}^p_{\geq 0}$, it has a minimal element.

Moreover the order is clearly compatible with the sum, this means that, given $\alpha = (a_1, \ldots, a_p)$ and $\beta = (b_1, \ldots, b_p)$ such that $\underline{a} \geq \underline{b}$ then for every $\gamma = (c_1, \ldots, c_p) \in \mathbb{Z}^p_{\geq 0}$, we have $\alpha + \gamma \geq \beta + \gamma$.

2.1.1 Homogeneous systems

Let's consider the case of a homogeneous system:

$$\underline{v}C \quad = \quad \underline{0}. \tag{2.2}$$

The set of solutions of the homogeneous system (2.2) is a sub-semigroup of $\mathbb{Z}^p_{\geq 0}$ and we denote it by $\mathrm{Sol}(\underline{v}C = \underline{0})$.

We want to show that $\mathrm{Sol}(\underline{v}C = \underline{0})$ is a finitely generated semigroup.

Theorem 2.1.2 ([CP], Theorem 9.18 on page 129)**.** *Let $A \subset \mathbb{Z}^p_{\geq 0}$ be a subset. Then the set M of all minimal elements of A is finite. Moreover, if $\alpha \in A$, then there exists $\nu \in M$ such that $\nu \leq \alpha$.*

We report it because it explains how we find minimal elements of A.

Proof. We proceed by induction on p. The statement clearly holds for $p = 1$. Suppose that the statement holds in $\mathbb{Z}_{\geq 0}^{p-1}$.

Let $j \in \{1, \dots, p\}$. We consider the projection of A on the j-th component:

$$\overline{A}_j = \{z \in \mathbb{Z}_{\geq 0} \mid z \text{ is the } j\text{−th component of some } \underline{a} \in A\},$$

and let

$$l_j = \min\{z \mid z \in \overline{A}_j \}.$$

Denote by L_j the set of all elements of A with j-th component equal to l_j:

$$L_j = \{\underline{a} \in A \mid a_j = l_j \}.$$

There is a bijection between L_j and the set:

$$L'_j = \{(a_1, \dots, \widehat{a}_j, \dots, a_p) \mid (a_1, \dots, a_p) \in L_j \}.$$

Hence, by induction, the set K_j of all minimal elements of L_j is finite.

Set $K = K_1 \cup \ldots \cup K_p$. K is finite. Let

$$k_j = \max\{a_j \mid \underline{a} \in K \} \quad \forall j = 1, \dots, p,$$

and define $\mu = (k_1, \dots, k_p)$. Then it is obvious that every element $\underline{k} \in K$ is such that $\underline{k} \leq \mu$.

Denote by M_j the set of all minimal elements of A with j-th component p_j satisfying $l_j \leq p_j \leq k_j$. From the inductive hypothesis, it follows that each set M_j is finite. Let $M = M_1 \cup \ldots \cup M_p$, then M is a finite set. Moreover, M is the set of all minimal elements of A.

Let $\gamma = (c_1, \dots, c_p)$ be any minimal element of A. If $\gamma \notin M$, then $\gamma \notin M_j$ for every $j = 1, \dots, p$. By definition of M_j and of l_j, we have $c_j > k_j$ for $j = 1, \dots, p$. Thus $\gamma > \mu$ and each element of K is consequently less then γ, contrary to the choice of γ as a minimal element of A.

The final assertion of the theorem follows because it is clear that any strictly descending chain of elements of A is finite. □

To conclude, we observe that $\mathbb{Z}_{\geq 0}^p$ is contained in the free additive abelian group $\mathbb{Z}^p$ and that the partial order on $\mathbb{Z}_{\geq 0}^p$ extends naturally on $\mathbb{Z}^p$ defining that:

$$\alpha \geq \beta \Longleftrightarrow \alpha - \beta \in \mathbb{Z}_{\geq 0}^p$$

Proposition 2.1.3 ([CP], Corollary 9.19 on page 130)**.** *Let G be a subgroup of $\mathbb{Z}^p$ and suppose that the set $S = G \cap \mathbb{Z}_{\geq 0}^p$ contains at least one non-zero element. Then S is a finitely generated sub-semigroup of $\mathbb{Z}_{\geq 0}^p$; in fact the set of minimal elements of S finitely generates S.*

Proof. Let M the set of minimal elements of S. Let $\alpha \in S \backslash \{\underline{0}\}$. Then there exists an element $\beta \in M$ such that $\alpha \geq \beta$. Then $\alpha - \beta \geq 0$ and so $\alpha - \beta \in \mathbb{Z}_{\geq 0}^p$. Since $\beta \in M$ and $\alpha \in S$ then $\alpha, \beta \in G$ and $\alpha - \beta \in G$ too. Consequently, either $\alpha = \beta$ or $\alpha - \beta \in S \backslash \{\underline{0}\}$. Hence, if $\alpha \neq \beta$ there exists $\beta_1 \in M$ such that $\beta_1 \leq \alpha - \beta$. It follows, as before, that $\alpha - \beta - \beta_1 \in S \backslash \{\underline{0}\}$ or $\alpha = \beta + \beta_1$. This process terminates in a finite number of steps because it determines a decreasing chain of elements in $S \backslash \{\underline{0}\}$ and it has always a minimal element (see Remark 2.1.1). At the end, the process gives: $\alpha = \beta + \beta_1 + \ldots + \beta_k$ and this shows that M generates the semigroup S. □

Applying these facts, we can prove that $\mathrm{Sol}(\underline{v}C = \underline{0})$ is finitely generated as a semigroup.

The zero-element $\underline{0}$ is always a solution of the system $\underline{v}C = \underline{0}$. Moreover, it is the minimum among the solutions of the system.

Proposition 2.1.4. $\mathrm{Sol}(\underline{v}C = \underline{0})$ *is finitely generated as sub-semigroup in* $\mathbb{Z}^p_{\geq 0}$.

Proof. If $\underline{0}$ is the unique solution of $\underline{v}C = \underline{0}$, then $\mathrm{Sol}(\underline{v}C = \underline{0}) = \{\underline{0}\}$ and it is finitely generated.

Suppose now that $\mathrm{Sol}(\underline{v}C = \underline{0}) \neq \{\underline{0}\}$, namely that the homogeneous system has a non-zero solution.

Let $\mathrm{Min}(\underline{v}C = \underline{0})$ be the set of minimal elements of $\mathrm{Sol}(\underline{v}C = \underline{0}) \setminus \{\underline{0}\}$. By Theorem 2.1.2 and Proposition 2.1.3, $\mathrm{Min}(\underline{v}C = \underline{0})$ is a finite set and generates $\mathrm{Sol}(\underline{v}C = \underline{0})$. □

We will always denote by $\mathrm{Min}(\underline{v}C = \underline{0})$ the set of minimal non-zero solutions of $\mathrm{Sol}(\underline{v}C = \underline{0})$. It is empty when $\mathrm{Sol}(\underline{v}C = \underline{0}) = \{\underline{0}\}$.

So every non-zero solution $\underline{a}$ of the system $\underline{v}C = \underline{0}$ is a $\mathbb{Z}_{\geq 0}$-linear combination of elements of $\mathrm{Min}(\underline{v}C = \underline{0})$:

$$\underline{a} = \sum_{i=1}^{t} \lambda_i \underline{\nu}_i$$

where $\lambda_i \in \mathbb{Z}_{\geq 0}$ and $\mathrm{Min}(\underline{v}C = \underline{0}) = \{\underline{\nu}_1, \ldots, \underline{\nu}_t\}$.

2.1.2 Non-homogeneous systems

Let's consider the case of non-homogeneous systems:

$$\underline{v}C = \underline{u}. \tag{2.3}$$

Computing the set of solutions and their representation is more complicated because the set of solutions is not a semigroup.

However, we can characterize the solutions using two sets: $\mathrm{Min}(\underline{v}C = \underline{0})$ and $\mathrm{Min}(\underline{v}C = \underline{u})$. $\mathrm{Min}(\underline{v}C = \underline{0})$ is the set of minimal non-zero solutions of the homogeneous system associated to $\underline{v}C = \underline{u}$, $\mathrm{Min}(\underline{v}C = \underline{u})$ is the set of minimal solutions of the system $\underline{v}C = \underline{u}$.

Moreover in this subsection we will suppose that $\mathrm{Min}(\underline{v}C = \underline{0}) \neq \emptyset$, hence $\mathrm{Sol}(\underline{v}C = \underline{0}) \neq \{\underline{0}\}$.

By Theorem 2.1.2 we have that $\mathrm{Min}(\underline{v}C = \underline{0})$ and $\mathrm{Min}(\underline{v}C = \underline{u})$ are finite subsets of $\mathbb{Z}^p_{\geq 0}$.

Remark 2.1.5. In [AC] and [CD] the authors describe $\mathrm{Min}(\underline{v}C = \underline{u})$ as the projection of a set in $\mathbb{Z}^{p+1}_{\geq 0}$, called M_1. M_1 is the set of elements of type $(\underline{m}, 1)$, where $(\underline{m}, 1)$ is a minimal solution of a system obtained introducing a new variable w:

$$\underline{v}'C' = \underline{0} \tag{2.4}$$

with $\underline{v}' = (\underline{v}, w)$ and C' is the matrix $C' = \begin{pmatrix} C \\ -\underline{u} \end{pmatrix}$.

The authors give this description for $\mathrm{Min}(\underline{v}C = \underline{u})$ because they want to build an efficient algorithm for solving systems of linear Diophantine equations.

Lemma 2.1.6. *Consider* $M_1 = \{(\underline{m}, 1) \mid (\underline{m}, 1) \in \mathrm{Min}(\underline{v}'C' = \underline{0})\} \subseteq \mathbb{Z}_{\geq 0}^{p+1}$ *and let* $\widetilde{M_1}$ *be the projection of* M_1 *in* $\mathbb{Z}_{\geq 0}^p$*. Then* $\mathrm{Min}(\underline{v}C = \underline{u}) = \widetilde{M_1}$.

Proof. Observe that for every $\underline{m} \in \widetilde{M_1}$ we have:

$$(\underline{m}, 1) \in \mathrm{Sol}(\underline{v}'C' = \underline{0}) \iff \underline{m} \in \mathrm{Sol}(\underline{v}C = \underline{u}).$$

Moreover

$$(\underline{m}, 1) \text{ is minimal in } \mathrm{Sol}(\underline{v}'C' = \underline{0}) \iff \underline{m} \text{ is minimal in } \mathrm{Sol}(\underline{v}C = \underline{u}).$$

Explicitly, if we suppose that $\underline{m}$ is not minimal, then there exists $\underline{s} \in \mathrm{Sol}(\underline{v}C = \underline{u})$ such that $\underline{s} < \underline{m}$. By definition, $(\underline{s}, 1) \in \mathrm{Sol}(\underline{v}'C' = \underline{0})$ and we have that $(\underline{s}, 1) < (\underline{m}, 1)$ because $\underline{s}_k < \underline{m}_k$ for each $k = 1, \ldots, p$ and the last coordinate is equal to 1. This contradicts that $(\underline{m}, 1) \in \mathrm{Min}(\underline{v}'C' = \underline{0})$.

Viceversa, we consider $\underline{s} \in \mathrm{Min}(\underline{v}C = \underline{u})$ and suppose that $(\underline{s}, 1) \notin M_1$. Then, there exists $(\underline{m}, 1) \in M_1$ such that $(\underline{m}, 1) \leq (\underline{s}, 1)$. Since the last component of $(\underline{m}, 1)$ and of $(\underline{s}, 1)$ is equal, then

$$(\underline{m}, 1) \leq (\underline{s}, 1) \iff \underline{m} \leq \underline{s}.$$

But this contradicts the assumption that $\underline{s} \in \mathrm{Min}(\underline{v}C = \underline{u})$. ☐

Remark 2.1.7. We observe that the element

$$\underline{\mu} + \sum_{i=1}^{t} \lambda_i \underline{\nu}_i$$

where $\underline{\mu} \in \mathrm{Min}(\underline{v}C = \underline{u})$, $\mathrm{Min}(\underline{v}C = \underline{0}) = \{\underline{\nu}_1, \ldots, \underline{\nu}_t\}$ and $\lambda_i \in \mathbb{Z}_{\geq 0}$, is always a solution of the system $\underline{v}C = \underline{u}$.

The converse is also true.

Proposition 2.1.8. *Any solution* $\underline{a} \in \mathbb{Z}_{\geq 0}^p$ *of the system* $\underline{v}C = \underline{u}$ *is the sum of an element of* $\mathrm{Min}(\underline{v}C = \underline{u})$ *and a* $\mathbb{Z}_{\geq 0}$*-linear combination of elements of* $\mathrm{Min}(\underline{v}C = \underline{0})$.

The proof of the proposition recalls the process used to find the solutions of a linear non-homogeneous system over a field (see [Se], [L]). The difference is that here we are working over the semigroup $\mathbb{Z}_{\geq 0}^p$ and not over a group.

Proof. Let $\underline{a}$ be a solution of the system (2.3). By definition of $\mathrm{Min}(\underline{v}C = \underline{u})$, we have

$$\exists\, \underline{\mu} \in \mathrm{Min}(\underline{v}C = \underline{u}) \quad \text{such that} \quad \underline{\mu} \leq \underline{a} \quad \text{and} \quad \underline{\mu}C = \underline{u}. \tag{2.5}$$

Consider the following element: $\underline{a} - \underline{\mu}$. Since $\underline{\mu} \leq \underline{a}$, then $\underline{a} - \underline{\mu} \in \mathbb{Z}_{\geq 0}^p$. Moreover

$$(\underline{a} - \underline{\mu})C = \underline{a}C - \underline{\mu}C = \underline{u} - \underline{u} \;=\; 0$$

that is, the element $\underline{a} - \underline{\mu}$ is a solution of the linear homogeneous system associated to the system (2.3). So $\underline{a} - \underline{\mu} \in \mathrm{Sol}(\underline{v}C = \underline{0})$. But $\mathrm{Sol}(\underline{v}C = \underline{0})$ is a semigroup finitely generated by $\mathrm{Min}(\underline{v}C = \underline{0})$, so if $\mathrm{Min}(\underline{v}C = \underline{0}) = \{\underline{\nu}_1, \dots, \underline{\nu}_k\}$, we have

$$\underline{a} - \underline{\mu} = \sum_{i=1}^{k} \lambda_i \underline{\nu}_i \quad \text{with } \lambda_i \in \mathbb{Z}_{\geq 0}$$

$$\underline{a} = \underline{\mu} + \sum_{i=1}^{k} \lambda_i \underline{\nu}_i \quad \text{with } \lambda_i \in \mathbb{Z}_{\geq 0}$$

and this proves the statement. □

2.2 Optimal solutions of a system of linear Diophantine equations

In this section we explain why we consider the systems of linear Diophantine equations.

Let X be a smooth and complete toric variety of dimension n. Fix an isomorphism $N \cong \mathbb{Z}^n$, so that we can think of elements in N as vectors in $\mathbb{Z}^n$. Suppose that we know the set of generators $G(\Sigma_X)$ and the list of maximal cones of Σ_X. Given a primitive collection $P = \{x_1, \dots, x_k\}$ of Σ_X, we want to compute its primitive relation $r(P)$.

By Proposition 1.3.5, we know that $G(\sigma_P) \subseteq G(\Sigma_X) \setminus P$. In order to find the set $G(\sigma_P)$ and the negative part of $r(P)$, we consider the linear system:

$$\sum_{y_i \in (G(\Sigma_X) \setminus P)} v_i y_i = x_1 + \dots + x_k \tag{2.6}$$

where the elements x_i and y_i are known, while the elements v_i are the unknowns of the system.

Rewrite (2.6) as $\underline{v}C = \underline{u}$, where $\underline{u} = x_1 + \dots + x_k$ and C is the matrix of coordinates of y_i, for $i = 1, \dots, \mathrm{card}(G(\Sigma_X) \setminus P)$. We refer to it as **the linear system associated to the primitive collection** P.

For every solution $\underline{a}$ of this linear system, we define

$$G(\underline{a}) = \{y_i \in (G(\Sigma_X) \setminus P) \mid a_i \neq 0\}.$$

This is the set of generators y_i whose coefficients in (2.6) are non-zero.

The following theorem characterizes a primitive relation using the linear system associated to P:

Theorem 2.2.1. *Let $P = \{x_1, \ldots, x_k\}$ be a primitive collection in $PC(X)$. Then the set of the coefficients of the negative part of $r(P)$ is a minimal solution of the system associated to P.*

Proof. Write:

$$r(P): \quad x_1 + \cdots + x_k = a_1 y_1 + \cdots + a_s y_s.$$

Moreover suppose that the system associated to P is:

$$\underline{v}C = \underline{u}.$$

Write $G(\Sigma_X) \setminus P = \{y_1, \ldots, y_s, y_{s+1}, \ldots, y_p\}$. The vector:

$$\underline{a} = (a_1, \ldots, a_p) \qquad \text{with } a_i = 0 \text{ for } i = s+1, \ldots, p$$

is a solution of the system associated to P.

Suppose that $\mathrm{Min}(\underline{v}C = \underline{u})$ is the set of minimal solutions of $\underline{v}C = \underline{u}$ and that $\mathrm{Min}(\underline{v}C = \underline{0}) = \{\underline{\nu}_1, \ldots, \underline{\nu}_t\}$ is the set of minimal non-zero solutions of the homogeneous system associated to $\underline{v}C = \underline{u}$.

By Proposition 2.1.8, we know that there exist $\lambda_1, \ldots, \lambda_t \in \mathbb{Z}_{\geq 0}$ and $\underline{\mu} \in \mathrm{Min}(\underline{v}C = \underline{u})$ such that:

$$\underline{a} \;=\; \underline{\mu} + \sum_{i=1}^{t} \lambda_i \underline{\nu}_i. \tag{2.7}$$

Let's show that $\lambda_i = 0 \; \forall \; i = 1, \ldots, t$.

Since $\underline{\mu}$ is minimal in $\mathrm{Sol}(\underline{v}C = \underline{u})$ we have $\underline{\mu} \leq \underline{a}$ and $\underline{\mu}C = \underline{u}$. Consider the set $G(\underline{\mu})$ and recall that $G(\sigma_P) = \{y_1, \ldots, y_s\}$. Then $G(\underline{\mu}) \subseteq G(\sigma_P)$ and we have:

$$\mu_1 y_1 + \cdots + \mu_s y_s = x_1 + \cdots + x_k.$$

Let H be the sub-group of N generated by $G(\sigma_P)$. Since X is smooth, $G(\sigma_P)$ is a basis for H. By definition $\underline{u} = x_1 + \cdots + x_k \in H$. Its coordinates with respect to this basis are uniquely determined, hence $\underline{a} = \underline{\mu}$ and $\lambda_i = 0$ for every $i = 1, \ldots, t$. □

Theorem 2.2.1 explains the importance of the set $\mathrm{Min}(\underline{v}C = \underline{u})$ associated to the system of linear Diophantine equations: it allows to find the primitive relation reducing the search to a finite set of $\mathbb{Z}^p_{\geq 0}$.

Once constructed $\underline{v}C = \underline{u}$, we are going to compute the set $\mathrm{Min}(\underline{v}C = \underline{u})$ of its minimal solutions. Since the cardinality of $\mathrm{Min}(\underline{v}C = \underline{u})$ can be greater than 1, we introduce:

Definition 2.2.2. *A minimal solution $(a_1, \ldots, a_p)$ is called an* ***optimal solution*** *if $G(\underline{a}) = \{y_i \mid a_i \neq 0\}$ is a set of generators of a cone of Σ_X.*

Then we can prove:

Proposition 2.2.3. *There exists a unique optimal solution and it gives the coefficients of the negative part of the primitive relation $r(P)$.*

Proof. By Theorem 2.2.1, the set of coefficients of the negative part of $r(P)$ is a minimal solution $\underline{a}$; clearly it is also optimal.

Suppose now that $\underline{b}$ is another optimal solution and define $\sigma_{\underline{b}} = \langle G(\underline{b})\rangle \in \Sigma_X$. Then

$$x_1 + \cdots + x_k = \sum_{y_i \in G(\underline{b})} b_i y_i$$

with all b_i's strictly positive, that is $x_1 + \cdots + x_k \in \mathrm{RelInt}(\sigma_{\underline{b}})$. The cone of Σ_X containing $x_1 + \cdots + x_k$ in its relative interior is unique, hence $\sigma_{\underline{b}} = \sigma_P$ and $G(\underline{b}) = G(\sigma_P) = G(\underline{a})$. Since the elements of $G(\sigma_P)$ are linearly independent, we get $\underline{a} = \underline{b}$. □

2.3 Commands in *Mathematica* computing minimal solutions of a linear systems of Diophantine equations

In this section we describe how we obtain the minimal solutions of a system of linear Diophantine equations $\underline{v}C = \underline{u}$ using the built-in function of *Mathematica* `Reduce` and the optimal solution of a linear system associated to a primitive collection P .

The command `Reduce` uses the algorithm implemented by Ajili and Contejean in [AC]. The command always returns a complete description of the set of solutions. It answers in the following form:

$$\{\texttt{C[1]}, \ldots, \texttt{C[h]} \in \texttt{Integers}, \texttt{C[1]} \geq 0, \ldots, \texttt{C[h]} \geq 0,$$

$$\underline{a}_1(\texttt{C[1]}, \ldots, \texttt{C[h]}), \ldots, \underline{a}_k(\texttt{C[1]}, \ldots, \texttt{C[h]})\,\}$$

Here `C[j]` are parameters in $\mathbb{Z}_{\geq 0}$, and $\underline{a}(\texttt{C[1]}, \ldots, \texttt{C[h]})$ are all solutions of the linear system, computed as explained in 2.1.2. (See [Wo]).

By replacing every parameter `C[j]` with zero, we find the k distinct minimal solutions of $\underline{v}C = \underline{u}$.

If $\underline{v}C = \underline{u}$ is the linear system associated to a primitive collection P of a variety X, then we use the command `Reduce` to compute the set of all minimal solutions $\mathrm{Min}(\underline{v}C = \underline{u})$.

Then for every minimal element $\underline{\mu} = (\mu_1, \ldots, \mu_p) \in \mathrm{Min}(\underline{v}C = \underline{u})$ we consider the set:

$$G(\underline{\mu}) = \{y_i \in \ (G(\Sigma_X) \setminus P) \mid \mu_i \neq 0\}.$$

The optimal solution is the solution corresponding to the unique set $G(\underline{\mu})$ which verifies the following condition:

$$\text{no primitive collection in } PC(X)\backslash\{P\} \text{ is contained in } G(\underline{\mu}). \tag{2.8}$$

Notice that the condition (2.8) is equivalent to say that $G(\underline{\mu})$ generates a cone in the fan of X.

In this way we can determine the primitive relation associated to P.

Remark 2.3.1. Observe that the command `Reduce` can have the following answers:

- 1-st: a unique solution: in this case $\mathrm{Min}(\underline{v}C = \underline{u})$ has cardinality 1 and $\mathrm{Min}(\underline{v}C = \underline{0})$ is empty;

- 2-nd: a unique solution depending from h positive integers parameters: in this case $\mathrm{Min}(\underline{v}C = \underline{u})$ of cardinality 1 and $\mathrm{Min}(\underline{v}C = \underline{0})$ has cardinality h;

- 3-rd: a set of k different solutions and no parameters, this means that $\mathrm{Min}(\underline{v}C = \underline{u})$ has cardinality k while $\mathrm{Min}(\underline{v}C = \underline{0})$ is empty;

- 4-nd: the more general case when there are k solutions written in terms of h parameters: in this case $\mathrm{Min}(\underline{v}C = \underline{u})$ has cardinality k and $\mathrm{Min}(\underline{v}C = \underline{0})$ has cardinality h.

Hence, in the case 1 the minimal solution gives the optimal solution directly. In the case 2 it is sufficient to replace the parameters `C[j]` with zero to obtain the optimal solution.

2.4 Algorithm computing a primitive relation

Let X be a smooth complete toric variety and suppose that its fan Σ_X is described by its maximal cones and its generators.

Moreover suppose that the set of all primitive collections $PC(X)$ is known and that we want to compute the primitive relation associated to a primitive collection $P \in PC(X)$.

Algorithm 2.4.1. PrimRel[X, PC(X)]

1. Let $P \in PC(X)$ be a primitive collection. *Mathematica* builds the linear system:

$$\sum_{y_i \in (G(\Sigma_X) \setminus P)} v_i y_i = \sum_{x \in P} x \tag{2.9}$$

   where x and y_i are known because they are defined by their coordinates, v_i are the unknowns of the system.

2. *Mathematica* solves the system using the command `Reduce`. Its answer is:

$$\{\texttt{C[1]}, \ldots, \texttt{C[h]} \in \texttt{Integers}, \texttt{C[1]} \geq 0, \ldots, \texttt{C[h]} \geq 0,$$

$$\underline{a}_1(\texttt{C[1]}, \ldots, \texttt{C[h]}), \ldots, \underline{a}_k(\texttt{C[1]}, \ldots, \texttt{C[h]})\}$$

3. *Mathematica* verifies if there are parameters which describe the solutions:
   (a) If there is no parameter, then it computes the cardinality of the set of solutions:
      i. if the cardinality equals one then the algorithm gives the primitive relation;
      ii. if the cardinality equals $k \geq 2$ then the algorithm has to compute the optimal solution:
         A. for every solution $\underline{a}_i$ computes the set $G(\underline{a}_i) = \{x_j \mid a_{ij} \neq 0\}$;
         B. it tests if $G(\underline{a}_i)$ generates a cone in Σ_X using the condition (2.8);

C. once found the set $G(\underline{a}_i)$ which generates a cone in Σ_X, the algorithm gives the primitive relation.

(b) If there are parameters, then *Mathematica* computes the set of minimal solutions replacing the parameters with 0.

(c) It computes the cardinality of the set of minimal solutions:

i. if the cardinality equals one then the algorithm gives the primitive relation;

ii. if the cardinality equals $k \geq 2$ then the algorithm has to compute the optimal solution:

A. for every solution $\underline{a}_i$ computes the set $G(\underline{a}_i) = \{x_j \mid a_{ij} \neq 0\}$;

B. it tests if $G(\underline{a}_i)$ generates a cone in Σ_X using the condition (2.8);

C. once found the set $G(\underline{a}_i)$ which generates a cone in Σ_X, the algorithm gives the primitive relation.

Chapter 3

Two ways to describe smooth complete toric varieties

In this chapter we introduce the way of describing a smooth complete toric variety X of dimension n and the algorithms to do it contained in the package *Toric Varieties*.

The chapter is divided in two sections: in the first we present how we can describe X: if X is a projective variety we will see that X has two equivalent representations, if X is not projective we do not know whether it is also true. We could give an affirmative answer understanding which link there is between the primitive relations of X and the $\mathbb{Z}$-module of syzygies among the generators of fan of X. In the second section we give the schemes of the algorithms.

3.1 Smooth complete toric varieties: algorithms describing them

Let X be a smooth complete toric variety of dimension n with fan Σ_X.

In Chapter 1 we introduced the concept of fan to describe a smooth complete toric variety because the combinatorial properties of Σ_X translate in a very efficient way the geometrical properties of X. Hence, the problem is how to introduce a smooth and complete toric variety using its fan.

The fan Σ_X is completely determined by its maximal cones and its generators, but when we study the Mori cone associated to the variety X it is very useful to know the primitive relations also.

So, we chose to represent Σ_X with a list where we can collect several information about X. The list is given by these five elements:

$$\{n, t, \mathrm{PR}, \mathrm{G}, \mathrm{MC}\} \tag{3.1}$$

where

- n is the dimension of variety;
- t is the cardinality of $G(\Sigma_X)$;

- PR is the set of all primitive relations;
- G is the set of generators, given as a list of elements of $\mathbb{Z}^n$;
- MC is the list of the maximal cones of Σ_X.

For a detailed description of the lists PR, MC and G see Section A.2.

With the previous notation, we can say that the fan Σ_X is completely determined by the data n, t, G and MC. Given these data, it is possible to compute all primitive collections and primitive relations of X. Computing the primitive collections is easy, in fact the primitive collections are the minimal subsets of $G(\Sigma_X)$ which are not contained in a cone of Σ_X. Then, for each primitive collection P, we compute the primitive relation $r(P)$ using the techniques described in Chapter 2. This involves solving a system of linear Diophantine equations.

On the other hand, suppose that we know the data: n, t and PR, i.e. we know the dimension of X, the number of generators of Σ_X and all primitive relations. One could ask whether these data determine X (up to isomorphism).

Clearly, the list of all primitive collections gives us the list of maximal cones in Σ_X: they are the subsets of $G(\Sigma_X)$ of cardinality n which do not contain a primitive collection. To complete the description, we would have to know the coordinates of generators of Σ_X with respect to a basis of N.

The question is: are these coordinates determined by all primitive relations?

This is equivalent to asking whether the set of primitive relations spans the vector space $\mathcal{N}_1(X)_\mathbb{Q}$ of linear relations among elements of $G(\Sigma_X)$.

The question has an affirmative answer in the projective case because if X is projective, the set of extremal classes generates all $\mathcal{N}_1(X)_\mathbb{Q}$ and the extremal classes are primitive relations. In fact, we have:

Proposition 3.1.1 ([Ba4], Theorem 2.2.4 - [Sa], Proposition 3.6)**.** *Let X_1, X_2 be two smooth, projective, toric varieties, and let Σ_1, Σ_2 be their fans in $N_\mathbb{Q}$. Then X_1 and X_2 are isomorphic if and only if there exists a bijection $\varphi : G(\Sigma_1) \to G(\Sigma_2)$ such that:*
for every $P = \{x_1, \dots, x_k\} \in PC(\Sigma_1)$ with primitive relation

$$r(P) : \quad x_1 + \cdots + x_k - (a_1 y_1 + \cdots + a_r y_r) = 0,$$

the relation

$$\varphi(r(P)) : \quad \varphi(x_1) + \cdots + \varphi(x_k) - (a_1 \varphi(y_1) + \cdots + a_r \varphi(y_r)) = 0$$

is a primitive relation for Σ_2 and this defines a bijection between the sets of primitive relations of the two fans.

Suppose that X is non projective and there exists an invariant subvariety V of X such that the blow-up $\mathrm{Bl}_V(X)$ of X along V is projective. In this case, it is known that the set of primitive relations generates all $\mathcal{N}_1(X)_\mathbb{Q}$ (see Corollary 5.6 in [C2]). Hence also in this case the question has a positive answer.

However, we were not able to give a positive answer to the question for any non-projective smooth complete toric variety, but we conjecture that there is an affirmative answer also in this case.

Let's consider the following example:

Example 3.1.2. Let $N \cong \mathbb{Z}^3$ and $N_{\mathbb{Q}} \cong \mathbb{Q}^3$. Let X_1 be the smooth, complete toric variety described in Example 1 in [FP], (see also the variety X_1 described in Section 7.2).

We denote by X the smooth complete toric variety of dimension 3, obtained by the blow-up of X_1 along the invariant curve $V(\langle$`x[2], x[5]`$\rangle)$. X is a non-projective variety and has not a projective blow-up along any toric subvariety V.

Let's consider the description of X given by its primitive relations (the letter `C` denotes the contractible primitive relations of X):

```
x[1] + x[5]          - x[2] - x[4]     =  0
x[1] + x[7]          - x[4]            =  0
x[1] + x[8]          - x[7]            =  0
x[1] + x[9]          -2 x[2] - x[4]    =  0  C
x[2] + x[5]          - x[9]            =  0  C
x[2] + x[6]          -2 x[3] -2 x[5]   =  0
x[2] + x[7]          - x[5]            =  0
x[2] + x[8]          - x[3] -2 x[5]    =  0
x[3] + x[4]          -2 x[1] - x[6]    =  0  C
x[3] + x[7]          - x[1] - x[6]     =  0
x[3] + x[8]          - x[6]            =  0
x[4] + x[8]          -2 x[7]           =  0  C
x[6] + x[9]          -2 x[3] -3 x[5]   =  0  C
x[7] + x[9]          -2 x[5]           =  0  C
x[8] + x[9]          - x[3] -3 x[5]    =  0  C
x[4] + x[5] + x[6]   - x[7] - x[8]     =  0
x[5] + x[6] + x[7]   -2 x[8]           =  0  C
```

In this case we can observe that the set $\{$`x[1],x[2],x[3]`$\}$ generates a maximal cone in the fan of X, so we fix the following identification:

$$\texttt{x[i]} -> e_i$$

where e_i denotes the i-th vector of the canonical basis of $\mathbb{Z}^3$.

Replacing `x[1]`, `x[2]`, `x[3]` with e_1, e_2, e_3 respectively in all primitive relations we can write the other generators as $\mathbb{Z}$-linear combination of the elements of the fixed basis of N:

$$\begin{aligned}
\texttt{x[4]} &= -e_2 - e_3 \\
\texttt{x[5]} &= -e_1 - e_3 \\
\texttt{x[6]} &= -2e_1 - e_2 \\
\texttt{x[7]} &= -e_1 - e_2 - e_3 \\
\texttt{x[8]} &= -2e_1 - e_2 - e_3 \\
\texttt{x[9]} &= -e_1 + e_2 - e_3
\end{aligned}$$

Hence, in this case the linear system given by all primitive relations determines completely the structure of the fan of the variety and the variety.

Since for projective varieties we have two equivalent representations and for non-projective varieties we conjecture that it should be true, we built some algorithms which present smooth complete toric varieties considering the possibility of describing them with both representations.

3.2 Algorithms introducing varieties

In this section we present the algorithms of the package to introduce and describe a smooth complete toric variety X. They are `NewVariety`, `AddInput[X]` and `TabVar[X]`.

The first allows to obtain one of the representations of a toric variety X. The program allows to the user to interact with *Mathematica*. In fact, it builds the list representing the variety X asking some simple questions to the user.

The algorithm `AddInput[X]` was built to collect information about X. For example, if X is given by the data n, t, G, MC, then it can be used to know the primitive relations of X.

Remark 3.2.1. Computing the list of primitive relations is always possible but it can request some time because the algorithm has to solve a system of linear Diophantine equations for every primitive collection. On the other hand, given the primitive relations, computing the coordinates of generators of Σ_X could not be possible. But, if we are able to do it, then we can see that the algorithm gives its answer very quickly.

The last algorithm allows to have an understandable description of primitive relations of X. In fact, it transforms every list representing a primitive relation (see Section A.2), in a linear polynomial.

Let's give the schemes of the three algorithms.

Algorithm 3.2.2. NewVariety.

1. *Mathematica* asks "Which is the dimension?"(of the variety)

2. *Mathematica* asks "Which is the the number of generators?"

3. *Mathematica* asks "Do you want to use either the primitive relation (p) or cones (c) to describe the variety?"
   If the user answers with `p` then

   (a) *Mathematica* says "Give primitive relations";
   (b) *Mathematica* transforms every primitive relation in the list of type (A.2) (see Section A.2);
   (c) *Mathematica* gives the output $\{n, t, \mathrm{PR}, \mathrm{Null}, \mathrm{Null}\}$.

   If the user answers with `c` then

   (a) *Mathematica* says "Give generators";
   (b) *Mathematica* says "Give maximal cones";
   (c) *Mathematica* gives the output $\{n, t, \mathrm{Null}, \mathrm{G}, \mathrm{MC}\}$.

Algorithm 3.2.3. AddInput[X].
The algorithm checks if X is described by:

1. its generators and maximal cones ($X = \{n, t, \mathrm{Null}, \mathrm{G}, \mathrm{MC}\}$), then *Mathematica* uses the algorithm `PrimRel` (see Chapter 2) and computes every primitive relation;

2. its primitive relations and generators ($X = \{n, t, \mathrm{PR}, \mathrm{G}, \mathrm{Null}\}$), then *Mathematica* computes all maximal cones;

3. its primitive relations ($X = \{n, t, \mathrm{PR}, \mathrm{Null}, \mathrm{Null}\}$), then:

   (a) *Mathematica* computes all maximal cones;

   (b) *Mathematica* verifies if the system of primitive relations admits a unique solution (it verifies if the rank of the matrix of coefficients of generators in the primitive relations has rank ρ);

      i. if the answer is `True` then it computes the coordinates of generators;

      ii. if the answer is `False` then it remarks the fact.

In every case the answer is the list

$$\{n, t, \mathrm{PR}, \mathrm{G}, \mathrm{MC}\}.$$

Algorithm 3.2.4. TabVar[X].

1. *Mathematica* transforms every primitive relation in a polynomial;

2. *Mathematica* determines all contractible primitive relations and puts beside them the letter `C`;

3. *Mathematica* answers with the table of primitive relations.

Chapter 4

Classes of curves

In this chapter we present the statements and the algorithms related to the classes of curves. Given a smooth complete toric variety X with fan Σ_X, a numerical class of 1-cycles is represented by a linear polynomial in the generators of Σ_X with integer coefficients.

Then we explain how we can use this polynomial description of a numerical class to check whether X is projective. It is connected with the Elimination theory and the problem of checking the existence of a solution of a system of linear Diophantine equations.

The same arguments are used to determine which extremal classes there are in the Mori cone of a projective smooth toric variety.

The chapter is divided in five sections: in the first we find the algorithms which compute the relation associated to numerical classes of invariant curves and the relation associated to contractible classes of invariant curves of a variety X. Section 4.2 is devoted to the statements which allow to compute the vector of coordinates of a numerical class of curves either in the set of all invariant curves or in the set of contractible classes (when X is projective). In the same section we find a brief description of the built-in function of *Mathematica* which we use to compute the coordinates of a numerical class of curves. Section 4.3 presents the problem of finding an equivalent computational condition for verifying whether a smooth and complete toric variety is projective and the description of the algorithm `ProjQ`. In the last section we consider a smooth projective toric variety X and prove Lemma 4.4.1. It represents a computational condition to verify whether a contractible class of X is extremal or not.

The references for toric Mori theory used throughout this chapter are [O2], [R], [Wi]. We refer to [AL], [CLO1], [CLO2] for Elimination theory and to [Wo] for the study of existence of a solution of a system of linear Diophantine equations.

4.1 Invariant curves and contractible classes

Let X be a smooth complete toric variety with fan Σ_X and generators $\{x_1, \ldots, x_t\}$.

As explained in Section 1.4, there is a natural isomorphism among $\mathcal{N}_1(X)$ (the group of numerical classes of 1-cycles in X) and the group of the linear relations among $x_1, \ldots, x_t$ in N.

In this section we present the algorithm to compute the relation associated to a numerical class of invariant curves.

Algorithm 4.1.1. Curve[X,cone].

1. Let σ be an $(n-1)$-dimensional cone in Σ_X generated by $x_1, \ldots, x_{n-1}$. It determines the two maximal cones σ' and σ'' such that $\sigma' \cap \sigma'' = \sigma$:

$$\begin{aligned} \sigma' &= \langle x_1, \ldots, x_{n-1}, x_n \rangle, \\ \sigma'' &= \langle x_1, \ldots, x_{n-1}, x_{n+1} \rangle. \end{aligned}$$

2. The algorithm builds the linear system on integers:

$$x_n + x_{n+1} = -\sum_{i=1}^{n+1} v_i x_i$$

   where v_i are unknowns.

3. It solves the system and finds the relation.

Remark 4.1.2. Observe that v_i are uniquely determined (Section 1.4).

Let X be a smooth complete toric variety described by the list PR of all primitive relations.

We want to give a criterion to determine which relations are contractible. It is the following:

Proposition 4.1.3 ([Sa], Theorem 4.10 - [C2], Proposition 3.4)**.** *Let $P = \{x_1, \ldots, x_h\}$ be a primitive collection in Σ_X, with primitive relation:*

$$r(P): \quad x_1 + \cdots + x_h - a_1 y_1 - \cdots - a_k y_k = 0.$$

Then $r(P)$ is contractible if and only if for every primitive collection Q of Σ_X such that $Q \cap P \neq \emptyset$ and $Q \neq P$, the set $(Q \setminus P) \cup \{y_1, \ldots, y_k\}$ contains a primitive collection.

Algorithm 4.1.4. ContractibleRel[X].

1. It applies Proposition 4.1.3;

2. It gives the list of all contractible primitive relations.

Observe that it is sufficient to know the set $PC(X)$.

In the appendix, we find three different algorithms which compute the list of contractible relations, and give the output in different forms. One builds a table where the primitive relations are given by linear polynomials in the generators, one answers with the list of all contractible primitive relations, where each is written as a linear polynomial in the generators. The last one gives a list where the primitive relations have the form (A.2) (see Section A.2).

4.2 The Mori cone

The section is devoted to show how we can describe $\mathcal{N}_1(X)$ fixing a basis B. We do it using the properties of Mori cone $\mathrm{NE}(X)$.

Observe that a subset $\mathcal{S} \subset \mathrm{NE}(X)_{\mathbb{Z}}$, which generates $\mathrm{NE}(X)_{\mathbb{Z}}$ as a semigroup, also generates $\mathcal{N}_1(X)$ as a group because $\dim \mathrm{NE}(X) = \rho_X$. Then the subset $\mathcal{S}$ has to contain a basis of $\mathcal{N}_1(X)$ (as a $\mathbb{Z}$-module).

We apply this fact in two cases:

1) when $\mathcal{S}$ is the set of all numerical classes of invariant curves, in this case we denote it by $\mathcal{I}$ (see Theorem 1.4.1);

2) when X is projective and $\mathcal{S}$ is the set of all contractible classes: in this case we denote it by $\mathcal{C}$ (see Theorem 1.5.3).

Recall from Section 1.4 that every $\gamma \in \mathcal{N}_1(X)$ is represented by a linear polynomial in the generators of Σ_X. More precisely, write $G(\Sigma_X) = \{x_1, \ldots, x_t\}$. Then the assignment

$$\gamma \mapsto \sum_{i=1}^{t} (\gamma \cdot V(x_i)) x_i$$

gives injective homomorphisms $\mathcal{N}_1(X) \hookrightarrow \mathbb{Z}[x_1, \ldots, x_t]$ and $\mathcal{N}_1(X)_{\mathbb{Q}} \hookrightarrow \mathbb{Q}[x_1, \ldots, x_t]$.

Let $\mathcal{S} = \{\gamma_1, \ldots, \gamma_d\}$ be a subset of $\mathrm{NE}(X)_{\mathbb{Z}}$ which generates $\mathrm{NE}(X)_{\mathbb{Z}}$ as a semigroup as in 1) or 2). We want to extract from $\mathcal{S}$ a basis of $\mathcal{N}_1(X)$.

Let $p_1, \ldots, p_d \in \mathbb{Z}[x_1, \ldots, x_t]$ be the relations associated to $\gamma_1, \ldots, \gamma_d$. Our goal is to write $d - \rho_X$ of the p_i's as an integral linear combination of the remaining ρ_X polynomials. In order to do this, we compute the syzygies among $p_1, \ldots, p_d$ in $\mathbb{Z}[x_1, \ldots, x_t]$. This means that we consider the ring of polynomials $\mathbb{Z}[f_1, \ldots, f_d]$ and define the following ring homomorphism:

$$\begin{array}{cccc} \phi: & \mathbb{Z}[f_1, \ldots, f_d] & \to & \mathbb{Z}[x_1, \ldots, x_t] \\ & f_i & \mapsto & p_i \qquad \text{for } i = 1, \ldots, d. \end{array}$$

Then $\mathrm{Ker}(\phi)$ is an ideal in $\mathbb{Z}[f_1, \ldots, f_d]$ and it is the set of all polynomials $g(f_1, \ldots, f_d)$ such that $g(p_1, \ldots, p_d) = 0$. These are the syzygies among $p_1, \ldots, p_d$: $\mathrm{Ker}(\phi) = \mathrm{Syz}(p_1, \ldots, p_d)$. It is enough to determine a system of generators of $\mathrm{Ker}(\phi)$ to describe $\mathrm{Syz}(p_1, \ldots, p_d)$ completely.

This problem can be solved working in the ring $\mathbb{Z}[f_1, \ldots, f_d, x_1, \ldots, x_t]$ and using Elimination theory. We have:

Proposition 4.2.1 ([AL] Theorem 2.4.2 page 81 - [CLO1], Theorem 2 page 113)**.** *Let $\{p_1, \ldots, p_d\}$ be a set of polynomials in $\mathbb{Z}[x_1, \ldots, x_t]$. Let $\mathcal{F}$ be the ideal generated by the polynomials $f_1 - p_1, \ldots, f_d - p_d$ in $\mathbb{Z}[f_1, \ldots, f_d, x_1, \ldots, x_t]$.*

Then

$$\mathrm{Syz}(p_1, \ldots, p_d) = \mathcal{F} \cap \mathbb{Z}[f_1, \ldots, f_d].$$

Proposition 4.2.1 gives an algorithm for computing a system of generators of $\mathrm{Syz}(p_1, \ldots, p_d)$. In fact, it is sufficient to compute a Gröbner basis $\mathcal{G}$ for the ideal $\mathcal{F}$ with respect to an elimination order in which the variables $x_1, \ldots, x_t$ are larger than the variables $f_1, \ldots, f_d$. Then the polynomials in $\widetilde{\mathcal{G}} = \mathcal{G} \cap \mathbb{Z}[f_1, \ldots, f_d]$ form a Gröbner basis for $\mathrm{Syz}(p_1, \ldots, p_d)$ and hence a system of generators for $\mathrm{Syz}(p_1, \ldots, p_d)$ (see [CLO1] Definition 5 on page 74 for Gröbner bases, and Definition 1 on page 113 for the elimination order).

In our case, the polynomials $p_1, \ldots, p_d$ are linear polynomials in the variables $x_1, \ldots, x_t$ and we can compute a Gröbner basis for $\mathrm{Syz}(p_1, \ldots, p_d)$ using Elimination theory for linear polynomials. It says that the Gröbner basis $\mathcal{G}$ can be computed by applying Gauss elimination to the matrix of coefficients of the polynomials $f_1 - p_1, \ldots, f_d - p_d \in \mathbb{Z}[f_1, \ldots, f_d, x_1, \ldots, x_t]$ (see [CLO1] Chapter 2).

So we consider the matrix $\mathcal{A}$ of coefficients of polynomials $f_1 - p_1, \ldots, f_d - p_d$ and apply Gauss elimination to $\mathcal{A}$. In this way, we obtain a new matrix $\mathcal{B}$ which is the matrix of coefficients of the elements of the Gröbner basis $\mathcal{G}$ of the ideal $\mathcal{F}$. The polynomials in $\widetilde{\mathcal{G}}$ give a Gröbner basis for $\mathrm{Syz}(p_1, \ldots, p_d)$.

Notice that the elements in $\widetilde{\mathcal{G}}$ are exactly $d - \rho_X$, because $\mathcal{N}_1(X)_{\mathbb{Q}}$ has dimension ρ_X and the set $\mathcal{S}$ contains a basis of $\mathcal{N}_1(X)$.

Once determined $\widetilde{\mathcal{G}}$ we can find a basis of $\mathcal{N}_1(X)$ contained in $\mathcal{S}$. In fact by construction, up to reordering the variables f_i, we can deduce from $\widetilde{\mathcal{G}}$ some syzygies of the form:

$$\begin{array}{l} f_{\rho_{X+1}} - \sum_{i=1}^{\rho_X} a_i^{\rho_{X+1}} f_i \\ \vdots \\ f_d - \sum_{i=1}^{\rho_X} a_i^d f_i \end{array}$$

Then by definition of syzygy we have:

$$\begin{array}{lcl} p_{\rho_{X+1}} & = & \sum_{i=1}^{\rho_X} a_i^{\rho_{X+1}} p_i \\ \vdots & & \\ p_d & = & \sum_{i=1}^{\rho_X} a_i^d p_i \end{array} \tag{4.1}$$

This means that $\{\gamma_1, \ldots, \gamma_{\rho_X}\}$ is a basis of $\mathcal{N}_1(X)$.

Moreover the equations (4.1) give the coordinates of $\gamma_{\rho_{X+1}}, \ldots, \gamma_d$ with respect to this basis.

This procedure is used to write the algorithms `EsprVett[X,set]`, where `X` is the variety and `set` is either the set $\mathcal{C}$ or $\mathcal{I}$, and `CoordClassesCurves[X]`, `CoordContrClasses[X]` where `X` is the variety.

4.2.1 Commands in *Mathematica* computing a basis B of $\mathcal{N}_1(X)$

In this section we introduce the built-in function `Eliminate` which allows us to find a basis B of $\mathcal{N}_1(X)$.

We use the same notations introduced in the previous section.

Let $\mathcal{S} = \{\gamma_1, \ldots, \gamma_d\}$. Every γ_i is represented by a polynomial p_i in $\mathbb{Z}[x_1, \ldots, x_t]$.

The command `Eliminate` computes a system of generators for $\mathrm{Syz}(p_1, \ldots, p_d)$.

The input of `Eliminate` are the polynomials $f_1-p_1, \ldots, f_d-p_d$ in $\mathbb{Z}[f_1, \ldots, f_d, x_1, \ldots, x_t]$, which define the ideal $\mathcal{F}$. The command eliminates the variables $x_1, \ldots, x_t$ using Elimination theory for linear polynomials. Let's analyze how it proceeds:

1. it computes the matrix $\mathcal{A}$ of coefficients of polynomials $f_i - p_i$ for $i = 1, \ldots, d$;
2. it applies Gauss elimination to $\mathcal{A}$ and determines the matrix $\mathcal{B}$ of coefficients of a Gröbner basis $\mathcal{G}$ of $\mathcal{F}$ with respect to the elimination order;
3. it answers giving the polynomials $g \in \mathcal{G} \cap \mathbb{Z}[f_1, \ldots, f_d]$.

Given the polynomials $g \in \mathcal{G} \cap \mathbb{Z}[f_1, \ldots, f_d]$, we wrote some algorithms

1. to obtain the equations (4.1);
2. to obtain the set $\mathcal{V}$ of vectors $\underline{w}_i$ of the coordinates of every $\gamma_i \in \mathcal{S}$ after the identification of $\mathcal{N}_1(X)$ with $\mathbb{Z}^{\rho_X}$.

4.2.2 Algorithm computing the coordinates of a numerical class of invariant curves

Algorithm 4.2.2. EsprVett[X,$\mathcal{S}$].

1. The algorithm builds the polynomials $f_i - p_i$ for $i = 1, \ldots, d = \mathrm{card}(\mathcal{S})$.
2. The algorithm uses the command `Eliminate` to compute a system of generators of $\mathrm{Syz}(p_1, \ldots, p_d)$: $\widetilde{\mathcal{G}}$.
3. The algorithm uses $\widetilde{\mathcal{G}}$ to compute a basis B of $\mathcal{N}_1(X)$.
4. It answers giving the set of vectors $\underline{w}_j$ of coordinates of all numerical classes in $\mathcal{S}$ with respect to the basis B:
$$\mathcal{V} = \{\underline{w}_1 = e_1, \ldots, \underline{w}_{\rho_X} = e_{\rho_X}, \underline{w}_{\rho_X+1}, \ldots, \underline{w}_d\},$$
where $\{e_1, \ldots, e_{\rho_X}\}$ denotes the canonical basis of $\mathbb{Q}^{\rho_X}$.

Remark 4.2.3. The other two algorithms which compute the vectors of the coordinates of numerical classes, `CoordContrClasses[X]` and `CoordCurvesClasses[X]`, follow the same scheme but they give their answer as a table. The first works only with projective varieties and considers the set $\mathcal{C}$, the second works with any variety and considers the set $\mathcal{I}$.

4.3 Projectivity

In Chapter 1 we considered the problem of verifying the projectivity of a smooth complete toric variety X and we have seen that it is equivalent to show that $\mathrm{NE}(X)$ is strongly convex.

By definition, $\mathrm{NE}(X)$ is strongly convex if and only if $\mathrm{NE}(X) \cap -\mathrm{NE}(X) = \{0\}$. Theorem 1.4.1 says $\mathrm{NE}(X)_{\mathbb{Z}}$ is generated by the set of numerical classes of invariant curves $\mathcal{I}$. Suppose that $\mathcal{I} = \{\gamma_1, \ldots, \gamma_d\}$.

Using the algorithm introduced in the previous subsection, we compute a basis B of $\mathcal{N}_1(X)$, contained in $\mathcal{I}$, and then the coordinates of all γ_i with respect to B. Hence, we have the set

$$\mathcal{V} = \{\underline{w}_i \mid i = 1, \ldots, d\}$$

which contains the canonical basis of $\mathbb{Q}^{\rho_X}$.

The vectors $\underline{w}_i$ allow to characterize the projectivity of X. We identify every class with the vector of its coordinates, such that $\mathrm{NE}(X)_{\mathbb{Z}} = \langle \underline{w}_1, \ldots, \underline{w}_d \rangle$.

Lemma 4.3.1. *The condition* $\mathrm{NE}(X) \cap -\mathrm{NE}(X) = \{0\}$ *is equivalent to verify that the equation*

$$\sum_{i=1}^{d} v_i \underline{w}_i = 0 \tag{4.2}$$

has only the trivial solution in $\mathbb{Z}^d_{\geq 0}$.

Proof. Suppose that there exists $\underline{w} \in \mathrm{NE}(X) \cap -\mathrm{NE}(X)$. This means that:

$$\begin{aligned} \underline{w} &= \sum_{i=1}^{d} a_i \underline{w}_i, \\ \underline{w} &= \sum_{i=1}^{d} b_i(-\underline{w}_i), \end{aligned}$$

where $a_i, b_i \in \mathbb{Z}_{\geq 0}$. Hence

$$\sum_{i=1}^{d} (a_i + b_i)\underline{w}_i = 0,$$

then $(a_1 + b_1, \ldots, a_d + b_d)$ is a non-trivial solution for (4.2), a contradiction.

Conversely, let $\underline{w}$ be an element in $\mathrm{NE}(X) \cap -\mathrm{NE}(X)$. We are going to prove that $\underline{w} = \underline{0}$. Since $\underline{w} \in \mathrm{NE}(X) \cap -\mathrm{NE}(X)$ then

$$\begin{aligned} \underline{w} &= \sum_{i=1}^{d} a_i \underline{w}_i, \\ \underline{w} &= \sum_{i=1}^{d} b_i(-\underline{w}_i), \end{aligned}$$

where $a_i, b_i \in \mathbb{Z}_{\geq 0}$, then

$$\sum_{i=1}^{d}(a_i + b_i)\underline{w}_i \quad = \quad 0. \tag{4.3}$$

By hypothesis, the equation (4.3) has only the trivial solution, hence $(a_i + b_i) = 0$ for all $i = 1, \dots, d$. This implies that $a_i = b_i = 0$ for all $i = 1, \dots, d$ and that $\underline{w} = \underline{0}$. □

Geometrically, Lemma 4.3.1 says that X is projective if and only if there are no effective invariant 1-cycles which are numerically equivalent to zero.

In order to determine the projectivity we need to test the existence of non-trivial solutions of a system of linear Diophantine equations. Since this can take a long time, we observe that in some cases we can get an answer directly.

Proposition 4.3.2 ([O1], Proposition 8.1 page 51)**.** *Every smooth complete toric surface is projective.*

Proposition 4.3.3 ([KlSt])**.** *A smooth complete toric variety X of dimension n and with Picard number $\rho_X \leq 3$ is projective.*

Proposition 4.3.4 ([Ba4], Proposition 3.2)**.** *Let X be a smooth complete toric variety and let* PR *be the list of all primitive relations. If the negative part of every primitive relation in* PR *is not zero, then X is not projective.*

Using also these three propositions we have written the algorithm `ProjQ[X]`. It works in the following way:

Algorithm 4.3.5. ProjQ[X].

1. The algorithm tests if X has dimension $n \leq 2$. If it is verified then the output is `True`, otherwise it proceeds with the next test.

2. The algorithm tests if X has $\rho_X \leq 3$. If it is verified then the output is `True`, otherwise it proceeds with the next test.

3. The algorithm tests if X is described by primitive relation and then tests if X verifies Proposition 4.3.4, then the output is `False`, otherwise it proceeds with the next test. (If X is described by the data of maximal cones and generators the algorithm overlooks this step).

4. The algorithm computes the set $\mathcal{I}$ of all relations associated to all different numerical classes of invariant curves with the algorithm `AllInvCurves[X]` (see Section A.13.6).

5. The algorithm computes a basis B of $\mathcal{N}_1(X)$ and the vectors of the coordinates with respect to B of all $\gamma_i \in \mathcal{I}$ with the algorithm `EsprVett[X,`$\mathcal{I}$`]`.

6. Given the set $\mathcal{V} = \{\underline{w}_i \mid i = 1, \dots, d = \mathrm{card}(\mathcal{I})\}$ of vectors of coordinates of all numerical classes in $\mathcal{I}$, the algorithm builds the equation:

$$\sum_{i=1}^{d} v_i \underline{w}_i = 0,$$

and verifies whether it has a non-trivial solution. It uses `FullSimplify[Exists[-,-]]`, (where in the first − we set the unknowns of the system and in the second the equation). If there is a non-trivial solution then the output is `False` otherwise it is `True`.

4.4 Extremal classes

In this section we consider a smooth projective toric variety X. In this case, we know that the set $\mathcal{C}$ of all contractible classes generates $\mathrm{NE}(X)_{\mathbb{Z}}$ (see Section 1.5). We also know that the set $\mathcal{E}$ of all extremal classes is a subset of $\mathcal{C}$, and we can easily compute $\mathcal{C}$ as explained in Section 4.1. We will now explain how one can determine $\mathcal{E}$ inside $\mathcal{C}$.

We recall the definition of a non-extremal class:

A contractible class γ is non-extremal if and only if there exists $Z_1, Z_2 \in \mathrm{NE}(X)$ such that $Z_1 + Z_2 \in \mathbb{Q}_{\geq 0}\gamma$ and $Z_1 \notin \mathbb{Q}_{\geq 0}\gamma$.

To build the algorithm we use the following:

Lemma 4.4.1. *Let X be a projective variety. Let γ be a contractible class and let $\mathcal{C} = \{\gamma_1 = \gamma, \ldots, \gamma_s\}$ be the set of all contractible classes in X. Then the following are equivalent:*

i) γ is not extremal;

ii) there exist $m_2, \ldots, m_s \in \mathbb{Q}_{\geq 0}$ such that

$$\gamma = \sum_{i=2}^{s} m_i \gamma_i. \tag{4.4}$$

Proof. $ii) \implies i)$
Since $\gamma \neq 0$ at least one m_i is non-zero; we can assume that $m_2 \neq 0$. We set

$$Z_1 = m_2\gamma_2 \quad \text{and} \quad Z_2 = \sum_{i=3}^{s} m_i\gamma_i.$$

Then, $Z_1 + Z_2 \in \mathbb{Q}_{\geq 0}\gamma$.

Suppose that $Z_1 \in \mathbb{Q}_{\geq 0}\gamma$. Then there exists $\lambda \in \mathbb{Q}_{\geq 0}$ such that $m_2\gamma_2 = \lambda\gamma$. Since every contractible class is primitive, this implies that $\gamma = \gamma_2$. Contradiction.

$i) \implies ii)$
Let γ be a non-extremal contractible class. Let $Z_1, Z_2 \in \mathrm{NE}(X)$ be such that $Z_1 + Z_2 \in \mathbb{Q}_{\geq 0}\gamma$ and $Z_1 \notin \mathbb{Q}_{>0}\gamma$. Then

$$Z_1 + Z_2 = \lambda\gamma, \tag{4.5}$$

with $\lambda \in \mathbb{Q}_{>0}$.

Since

$$\mathrm{NE}(X) = \sum_{\eta \in \mathcal{E}} \mathbb{Q}_{\geq 0}\eta$$

and γ is non-extremal, we have

$$Z_i = \sum_{j=2}^{s} a_j^i \gamma_j$$

where $i = 1, 2$, $a_j^i \in \mathbb{Q}_{\geq 0}$. Then

$$\begin{aligned} Z_1 + Z_2 &= \sum_{i=1}^{2} \sum_{j=2}^{s} a_j^i \gamma_j \\ &= \lambda\gamma. \end{aligned}$$

Since $\lambda \neq 0$, it follows that:

$$\gamma = \sum_{j=2}^{s} \left(\frac{a_j^1 + a_j^2}{\lambda} \right) \gamma_j \tag{4.6}$$

and we have the statement. □

We use Lemma 4.4.1 to build the algorithm `ExtremalClasses[X]`.

Algorithm 4.4.2. ExtremalClasses[X].

1. The algorithm computes the set of all contractible classes $\mathcal{C}$.
2. The algorithm computes the set $\mathcal{V} = \{\underline{w}_i \mid i = 1, \ldots, s\}$ of all coordinates of contractible classes in $\mathcal{C}$ with respect to a basis B of $\mathcal{N}_1(X)$.
3. For every contractible class $\gamma_i \in \mathcal{C}$ the algorithm builds the following equation:

$$\underline{w}_i = \sum_{j \neq i} v_i \underline{w}_j \tag{4.7}$$

   and verifies if (4.7) has a non-trivial solution. The index i for which the equation (4.7) has a non-trivial solution says us that the class γ_i is non-extremal.

Also in this case, we solve the problem considering a system of linear Diophantine equations. Again we use the command `FullSimplify[Exists[-, -]]`, to know if (4.7) has a non-trivial solution.

Chapter 5

Toric blow-ups

In this chapter we will present the algorithm to compute the blow-up of a smooth complete toric variety X along an invariant subvariety V. The new variety is again smooth, complete and toric. Let $Y = \mathrm{Bl}_V(X)$ be the blow-up of X along V.

The algorithm works with both the two descriptions of the variety X and it answers giving the new variety Y. The list, which describes Y, has the same form of the list representing X.

The main problem is the description of how the primitive relations change after the blow-up. In Theorem 4.3 in [Sa] Sato enumerates primitive collections of Y and says how they derive from primitive collections of X. In order to obtain all primitive relations of Y we will need the result of Chapter 2.

We compare our algorithm with the partial algorithm written by C. Casagrande in her Ph-D thesis [C1]. Her algorithm is uncomplete, as she asserts in the appendix of her Ph-D thesis [C1], but it is a first try to compute primitive relations of variety Y.

We present two examples in which her algorithm fails and we will try to explain how her program works and why fails.

We will call Casagrande's algorithm `CBlowup` and our algorithm `Blowup`.

This chapter is divided in following sections: in Section 5.1 we expose the construction of a blow-up describing the maximal cones of the fans of X and Y. We introduce Sato's Theorem and the techniques to write our algorithm. In Section 5.2, we explain the algorithm written by Casagrande and present it in two examples. In the last section we present our algorithm.

For the definitions of blow-up we refer to [E2], [O2] and [Sa]. For the statements of Section 5.1 we find detailed references in articles [Sa] and [C2].

5.1 The blow-up of smooth complete toric variety

Let X be a complete, smooth, toric variety of dimension n and let Σ_X be the associated fan.

Fix the cone $\sigma = \langle x_1, \ldots, x_r \rangle \in \Sigma_X$ and let $V = V(\sigma)$ the invariant subvariety corresponding to σ. We denote by $Y = \mathrm{Bl}_V X$ the blow-up of X along V and by Σ_Y its fan. Let $x = x_1 + \cdots + x_r$ be the new generator in Σ_Y. Moreover we use the notation $PC(X)$ and $PC(Y)$ for the sets of primitive collections of X and Y respectively.

The construction of the variety Y is equivalent to a star subdivision of the cone σ in Σ_X.

Definition 5.1.1. *Under the above assumptions, the star subdivision of Σ_X along σ is:*

$$\begin{aligned}\Sigma^* = \Sigma_Y \quad = \quad & \Big\{ \tau \in \Sigma_X \mid \sigma \nsubseteq \tau \Big\} \cup \\ & \Big\{ \sigma_i + \nu \mid \sigma + \nu \in \Sigma_X, \sigma \cap \nu = \{0\},\ i = 1, \ldots, r \Big\},\end{aligned}$$

where $\sigma_i = \langle x_1, \ldots, \widehat{x}_i, \ldots, x_r, x \rangle$.

We describe a star subdivision with a simple example:

Example 5.1.2. Let $N \cong \mathbb{Z}^3$ be. We defined the fan generated by:

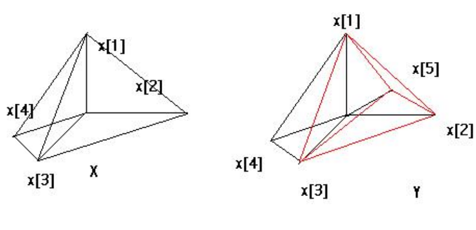
```
x[1] = (1,0,0), x[2] = (0,1,0), x[3] = (0,0,1), x[4] = (-1,-1,-1),
```

and its maximal cones are:

$\langle$x[1], x[2], x[3]$\rangle$, $\langle$x[1], x[2], x[4]$\rangle$,
$\langle$x[1], x[3], x[4]$\rangle$, $\langle$x[2], x[3], x[4]$\rangle$.

This fan determines $X = \mathbb{P}^3$. We blowup X in the (invariant) point $P = V(\langle$x[1],x[2],x[3]$\rangle)$. The fan of $Y = \mathrm{Bl}_P(X)$ is generated by $G(\Sigma_Y) = \{$x[1],x[2],x[3],x[4],x[5]$\}$, where the new generator is given by x[5] $=$ x[1] $+$ x[2] $+$ x[3] and its maximal cones are:

$\langle$x[1], x[2], x[4]$\rangle$, $\langle$x[1], x[3], x[4]$\rangle$, $\langle$x[2], x[3], x[4]$\rangle$,
$\langle$x[1], x[2], x[5]$\rangle$, $\langle$x[1], x[3], x[5]$\rangle$, $\langle$x[2], x[3], x[5]$\rangle$.



In the following theorem, Sato gives the relation among primitive collections of X and primitive collections of Y:

Theorem 5.1.3 ([Sa], Theorem 4.3). *The set of primitive collections of Y is given by:*

1. $G(\sigma) = \{x_1, \ldots, x_r\}$*;*
2. *every $P \in PC(X)$ such that $G(\sigma) \nsubseteq P$;*
3. *the minimal elements in the set $\{P' = (P \setminus G(\sigma)) \cup \{x\} \mid P \in PC(X),\ P \cap G(\sigma) \neq \emptyset\}$.*

We get the following:

Corollary 5.1.4. *Let $P \in PC(X)$.*

1. *If $P \cap G(\sigma) = \emptyset$, then P is a primitive collection for Σ_Y.*
2. *If $G(\sigma)$ is contained in P, then $P' = (P \setminus G(\sigma)) \cup \{x\}$ is a primitive collection for Σ_Y.*
3. *If $P \cap G(\sigma) \neq \emptyset$ and $G(\sigma) \nsubseteq P$ then P is a primitive collection in Σ_Y. Moreover the minimal elements in the set $\{P' = (P \setminus G(\sigma)) \cup \{x\} \mid P \in PC(X),\ P \cap G(\sigma) \neq \emptyset\ \}$ are primitive collections for Σ_Y.*

These three ways give all primitive collections of Σ_Y except $G(\sigma)$.

Proof. It is sufficient to prove that every set $P' = (P \setminus G(\sigma)) \cup \{x\}$ for every P containing $G(\sigma)$ is a minimal set in $\{P' = (P \setminus G(\sigma)) \cup \{x\} \mid P \in PC(X),\ P \cap G(\sigma) \neq \emptyset\}$.

Suppose that P' is not minimal, then there exists a primitive collection Q in $PC(X) \setminus \{P\}$ such that:

$$Q' = (Q \setminus G(\sigma)) \cup \{x\} \subset P',$$

then $Q \setminus G(\sigma) \subset P \setminus G(\sigma)$, hence $Q \subset P$. This is a contradiction. □

Observe that if we know Σ_X, we could first compute Σ_Y using Definition 5.1.1 and then compute directly the primitive collections of Y. However, given the primitive collections of X, Corollary 5.1.4 gives a more efficient way to compute $PC(Y)$.

We divide the primitive collection of Σ_X in three sets:

$$\begin{aligned} A &= \{P \in PC(X) \mid P \cap G(\sigma) = \emptyset\ \}, \\ B &= \{P \in PC(X) \mid G(\sigma) \subset P\}, \\ C &= \{P \in PC(X) \mid P \cap G(\sigma) \neq \emptyset \text{ and } G(\sigma) \nsubseteq P\}. \end{aligned}$$

Every element of A is a primitive collection for Y also. Elements of the set B determine primitive collections $P' = (P \setminus G(\sigma)) \cup \{x\}$. At last, consider the set C. The element of set C are primitive collections for Y, moreover these elements can determine other primitive collections. In fact, given the set:

$$C' = \{P' = (P \setminus G(\sigma)) \cup \{x\} \mid P \in B \cup C\ \},$$

the minimal elements P' of C' are primitive collections for Y.

For shortness, we call primitive collections of Y coming from collections in the sets A, B, C as primitive collections of type a), b), c), respectively. All minimal elements of the set C' which are not of type b) will be called primitive collections of type c').

We will denote $r^X(P)$ the primitive relation of primitive collection P, if $P \in PC(X)$ and $r^Y(P)$ if $P \in PC(Y)$. We will use a similar notation for the cone associated to a primitive collection: σ_P^X and σ_P^Y will denote the cone associated to $P \in PC(X)$ and the cone associated to $P \in PC(Y)$ respectively.

We are going to show that if one knows the primitive relations of X, it is possible to compute easily the relation associated to any primitive collection of type a), b), c) in Y. (See also [C2], Section 5).

Lemma 5.1.5. *Let $P \in A$ be hence $P \in PC(Y)$ is a primitive collection of type a).*

If $G(\sigma) \nsubseteq G(\sigma_P^X)$ then $r^Y(P) = r^X(P)$.

If $G(\sigma) \subseteq G(\sigma_P^X)$ let $r^X(P)$ be:

$$z_1 + \cdots + z_t - (a_1x_1 + \cdots + a_rx_r + b_{r+1}y_1 + \cdots + b_{r+s}y_s) = 0,$$

with $s \in \mathbb{Z}_{\geq 0}$. Let $m = \min\{a_1, \ldots a_r\}$ be. Then $r^Y(P)$ is

$$z_1 + \cdots + z_t - ((a_1 - m)x_1 + \cdots + (a_r - m)x_r + mx + b_{r+1}y_1 + \cdots + b_{r+s}y_s) = 0.$$

Proof. If $G(\sigma) \nsubseteq G(\sigma_P^X)$, then by Definition 5.1.1, $\sigma_P^X \in \Sigma_Y$, hence $G(\sigma_P^Y) = G(\sigma_P^X)$ and $r^Y(P) = r^X(P)$.

Suppose that $G(\sigma) \subseteq G(\sigma_P^X)$. Let $\{i_1, \ldots, i_k\}$ be the indices $i \in \{1, \ldots, r\}$ such that $a_i > m$. Then $k < r$. By Definition 5.1.1, the cone $\langle x, x_{i_1}, \ldots, x_{i_k}, y_1, \ldots, y_s\rangle \in \Sigma_Y$. Moreover we have:

$$z_1 + \cdots + z_t - ((a_{i_1} - m)x_{i_1} + \cdots + (a_{i_k} - m)x_{i_k} + mx + b_{r+1}y_1 + \cdots + b_{r+s}y_s) = 0.$$

So the above relation is $r^Y(P)$. □

Lemma 5.1.6. *Let $P \in B$ and let $P' \in PC(Y)$ be the primitive collection of type b) coming from P. Then $r^Y(P')$ is obtained by replacing the sum $x_1+\cdots+x_r$ with x in the positive part of primitive relation $r^X(P)$.*

Proof. Since $G(\sigma) \subseteq P$ then $G(\sigma_P^X) \cap G(\sigma) = \emptyset$ by Proposition 1.3.5. Suppose that

$$r^X(P): \quad x_1 + \cdots + x_r + x_{r+1} + \cdots + x_{r+s} - a_1y_1 - \cdots - a_ty_t = 0,$$

then $\langle y_1, \ldots, y_t\rangle \in \Sigma_Y$ and

$$r^Y(P'): \quad x + x_{r+1} + \cdots + x_{r+s} - a_1y_1 - \cdots - a_ty_t = 0$$

is the primitive relation associated to P'. □

Lemma 5.1.7. *Let $P \in C$ be. Then $P \in PC(Y)$ and its primitive relation $r^Y(P)$ is equal to $r^X(P)$.*

Proof. Since P is of type c) and by Proposition 1.3.5 then $G(\sigma) \not\subseteq G(\sigma_P)$, so $\sigma_P^X \in \Sigma_Y$. Hence $G(\sigma_P^Y) = G(\sigma_P^X)$ and $r^Y(P) = r^X(P)$. □

For primitive collections of type c') there is no known algorithm to compute $r^Y(P')$ from $r^X(P)$.

5.2 Casagrande's algorithm

Algorithm 5.2.1. CBlowup[X, cone].

1. The algorithm computes all primitive collections of Y, using Corollary 5.1.4.
2. Then it computes the primitive relations associated to primitive collections of types a), b), c), using Lemmas 5.1.5, 5.1.6 and 5.1.7.
3. It tries to compute the primitive relations of type c'). (We explain the proceeding of the algorithm afterward). If the algorithm is not able to compute them it answers `UNCOMPLETE`.

Let's analyze the proceeding used to compute the primitive relation of type c') and see when it fails.

We consider the primitive collection $P = \{z_1, \dots, z_k, z_{k+1}, \dots, z_t\}$, where $x_i = z_i$, for $i = 1, \dots, k$ which determines the minimal element $P' = \{z_{k+1}, \dots, z_t, x\}$.

The idea of Casagrande is to obtain the primitive relation of P' modifying the primitive relation of primitive collection P which determines P'. Suppose that

$$r^X(P) = r^Y(P) : \quad z_1 + \dots + z_t - a_1 y_1 - \dots - a_s y_s = 0, \tag{5.1}$$

then, she modifies this relation a first time considering:

$$z_{k+1} + \dots + z_t + x - a_1 y_1 - \dots - a_s y_s - x_{k+1} - \dots - x_r.$$

Then she sums the coefficients of same generators obtaining:

$$z_{k+1} + \dots + z_t + x - b_1 y_1 - \dots - b_p y_p.$$

At this point, she verifies if the generators with negative coefficient span a cone in Σ_Y: if the set of generators $\{y_1, \dots, y_p\}$ generates a cone in Σ_Y then she stops the proceeding, otherwise, she determines all primitive collections (could be more than one) which are contained in the negative part and modifies the negative part another time. The change consists in summing $mr^X(Q)$, where Q is a primitive collection such that $Q \subset \{y_1, \dots, y_p\}$ and $m = \min\{b_i \mid y_i \in Q\}$. Again, she verifies if the generators with negative coefficients span a cone in Σ_Y. If the check has an affirmative answer, then the algorithm collects the primitive relation, otherwise it writes `UNCOMPLETE` in the list which represents the variety.

Remark 5.2.2. The algorithm has been built to iterate the search of primitive collections contained in the modified negative part almost two times.

The proceeding explained above produces in some cases the right solution but we have two simple examples where the algorithm cannot compute the negative part of some primitive relations.

Example 5.2.3. Let X be the projective space of dimension 7 and consider the following blow-ups:

$$\begin{aligned} X_1 &= BL_{V_1}X \\ X_2 &= BL_{V_2}X_1 \\ X_3 &= BL_{V_3}X_2 \\ X_4 &= BL_{V_4}X_3 \\ X_5 &= BL_{V_5}X_4 \\ X_6 &= BL_{V_6}X_5 \\ X_7 &= BL_{V_7}X_6 \end{aligned}$$

where $V_1, \ldots, V_7$ are described by the following cones respectively

$$V_1 = V(\langle \texttt{x[1], x[2], x[3], x[4]} \rangle),$$

$$V_2 = V(\langle \texttt{x[1], x[2], x[5], x[6]} \rangle),$$

$$V_3 = V(\langle \texttt{x[1], x[2], x[4], x[6], x[7]} \rangle),$$

$$V_4 = V(\langle \texttt{x[4], x[5], x[7]} \rangle),$$

$$V_5 = V(\langle \texttt{x[4], x[5], x[8], x[9]} \rangle),$$

$$V_6 = V(\langle \texttt{x[4], x[9], x[10]} \rangle),$$

$$V_7 = V(\langle \texttt{x[4], x[9]} \rangle).$$

Now, consider X_7. It is given by these primitive relations (the letter `C` denotes the contractible primitive relations of X):

```
x[3] + x[11] - x[6] - x[7] - x[9]                                  = 0
x[3] + x[14] - x[5] - x[6] -2 x[9]                                 = 0
x[4] + x[9] - x[15]                                                = 0  C
x[5] + x[11] - x[4] - x[7] - x[10]                                 = 0
x[7] + x[13] - x[8] - x[9] - x[12]                                 = 0  C
x[10] + x[15] - x[14]                                              = 0  C
x[11] + x[12] -2 x[4] -2 x[7] - x[10]                              = 0  C
x[11] + x[13] - x[1] - x[2] -2 x[4]                                = 0
x[3] + x[4] + x[10] - x[5] - x[6] - x[9]                           = 0
x[3] + x[10] + x[12] -2 x[5] - x[6] - x[7] - x[9]                  = 0  C
x[3] + x[10] + x[13] -2 x[5] - x[6] - x[8] -2 x[9]                 = 0  C
x[4] + x[5] + x[7] - x[12]                                         = 0
x[5] + x[7] + x[14] - x[9] - x[10] - x[12]                         = 0  C
x[5] + x[7] + x[15] - x[9] - x[12]                                 = 0
x[5] + x[8] + x[14] - x[10] - x[13]                                = 0  C
x[5] + x[8] + x[15] - x[13]                                        = 0
x[6] + x[12] + x[13] -2 x[4] - x[5]                                = 0
x[7] + x[8] + x[14] - x[1] - x[2] - x[4]                           = 0
x[8] + x[11] + x[14] -2 x[1] -2 x[2] -2 x[4] - x[6]                = 0  C
x[8] + x[12] + x[14] - x[1] - x[2] -2 x[4] - x[5]                  = 0
x[10] + x[12] + x[13] - x[1] - x[2] -2 x[4] -2 x[5]                = 0
x[12] + x[13] + x[14] - x[1] - x[2] -2 x[4] -2 x[5] - x[15]        = 0  C
x[1] + x[2] + x[3] + x[4] - x[9]                                   = 0
x[1] + x[2] + x[3] + x[12] - x[5] - x[7] -  x[9]                   = 0
x[1] + x[2] + x[3] + x[13] - x[5] - x[8] -2 x[9]                   = 0
x[1] + x[2] + x[3] + x[15] -2 x[9]                                 = 0
x[1] + x[2] + x[5] + x[6] - x[10]                                  = 0
x[1] + x[2] + x[6] + x[12] - x[4] - x[7] - x[10]                   = 0
x[1] + x[2] + x[6] + x[13] - x[8] - x[14]                          = 0  C
x[6] + x[8] + x[9] + x[12] - x[4]                                  = 0
x[6] + x[8] + x[12] + x[15] -2 x[4]                                = 0
x[7] + x[8] + x[9] + x[10] - x[1] - x[2]                           = 0
x[8] + x[9] + x[10] + x[11] -2 x[1] -2 x[2]- x[4] - x[6]           = 0
x[8] + x[9] + x[10] + x[12] - x[1] - x[2] - x[4] - x[5]            = 0
x[1] + x[2] + x[4] + x[6] + x[7] - x[11]                           = 0
x[1] + x[2] + x[6] + x[7] + x[14] - x[9] - x[10] - x[11]           = 0  C
x[1] + x[2] + x[6] + x[7] + x[15] - x[9] - x[11]                   = 0
x[5] + x[6] + x[7] + x[8] + x[9]                                   = 0
```

We compute the blow-up of X_7 with respect to the variety V_8 determined by the cone $\langle$ `x[5], x[6], x[7], x[9]` $\rangle$. We have that:

set A	$P_1, P_2, P_6, P_7, P_8, P_9, P_{10}, P_{11}, P_{19}, P_{20}, P_{21}, P_{22}, P_{23}, P_{24}, P_{25}, P_{26}$
set B	P_{38}
set C	$P_3, P_4, P_5, P_{12}, P_{13}, P_{14}, P_{15}, P_{16}, P_{17}, P_{18}, P_{27}, P_{28}, P_{29}, P_{30}, P_{31}, P_{32}, P_{33}, P_{34}, P_{35}, P_{36}, P_{37}$

We consider and analyze the set A: the elements in A are primitive collections for X_8 and all have the same primitive relation associated except the primitive collection P_{10}. In fact, $\sigma_{P_{10}}$ contains the cone $\langle$`x[5], x[6], x[7], x[9]`$\rangle$. The primitive relation associated to P_{10} is:

$r^{X_8}(P_{10})$: `x[3] + x[10] + x[12] - x[5] - x[16]=0.`

Every primitive collection in C is a primitive collection for X_8 and their primitive relation associated does not change after the blow-up.

We compute the set C' of all sets $(P \setminus G(\sigma)) \cup \{$`x[16]`$\}$ for $P \in B \cup C$ and determine the minimal elements of C':

P'_3 = `{x[4], x[16]},`
P'_4 = `{x[11], x[16]},`
P'_5 = `{x[13], x[16]},`
P'_{12} = `{x[4], x[16]},`
P'_{13} = `{x[14], x[16]},`
P'_{14} = `{x[15], x[16]},`
P'_{15} = `{x[8], x[14], x[16]},`
P'_{16} = `{x[8], x[15], x[16]},`
P'_{17} = `{x[12], x[13], x[16]},`
P'_{18} = `{x[8], x[14], x[16]},`
P'_{27} = `{x[1], x[2], x[16]},`
P'_{28} = `{x[1], x[2], x[12], x[16]},`
P'_{29} = `{x[1], x[2], x[13], x[16]},`
P'_{30} = `{x[8], x[12], x[16]},`
P'_{31} = `{x[8], x[12], x[15], x[16]},`
P'_{32} = `{x[8], x[10], x[16]},`
P'_{33} = `{x[8], x[10], x[11], x[16]},`

$$
\begin{aligned}
P'_{34} &= \texttt{\{x[8], x[10], x[12], x[16]\}},\\
P'_{35} &= \texttt{\{x[1], x[2], x[4], x[16]\}},\\
P'_{36} &= \texttt{\{x[1], x[2], x[14], x[16]\}},\\
P'_{37} &= \texttt{\{x[1], x[2], x[14], x[15], x[16]\}},\\
P'_{38} &= \texttt{\{x[8], x[16]\}}.
\end{aligned}
$$

The primitive collection P'_{38} is of type b) and its primitive relation is:

$$r(P'_{38}): \quad \texttt{x[8] + x[16]=0.}$$

The minimal ones, which are not in B, are:

coll of type c')	$P'_3, P'_4, P'_5, P'_{13}$ P'_{14}, P'_{27}

The algorithm `CBlowup` is not able to compute the two primitive relations associated to the primitive collections $P'_4 =$ `{x[11],x[16]}` and $P'_5 =$ `{x[13],x[16]}`. It gives the output `UNCOMPLETE`.

This happens because the algorithm would have iterated the process to determine the negative part for three time not only twice.

Now we are going to present another example: in this case, there are two primitive collections whose primitive relation can not be computed with `CBlowup`, because the technique used by `CBlowup` generates an infinity cycle.

Example 5.2.4. Let T be the Oda 3-fold (see [O2] on page 84). It is described by the following primitive relations:

```
x[1] + x[4]          - x[7]          =  0
x[1] + x[5]          - x[2] - x[7]   =  0  C
x[2] + x[4]          - x[5]          =  0
x[2] + x[6]          - x[3] - x[5]   =  0  C
x[3] + x[4]          - x[6]          =  0
x[3] + x[7]          - x[1] - x[6]   =  0  C
x[5] + x[6] + x[7]   -2 x[4]         =  0  C
```

Consider the blow-up of T along the subvariety $V = V(\langle \texttt{x[2], x[7]} \rangle)$:

```
x[1] + x[4]          -x[7]        =  0
x[1] + x[5]          -x[8]        =  0  C
x[2] + x[4]          -x[5]        =  0
x[2] + x[6]          -x[3]-x[5]   =  0  C
x[2] + x[7]          -x[8]        =  0  C
x[3] + x[4]          -x[6]        =  0
x[3] + x[7]          -x[1]-x[6]   =  0  C
x[3] + x[8]                       =  0
x[4] + x[8]          -x[5]-x[7]   =  0  C
x[6] + x[8]          -x[4]        =  0
x[5] + x[6] + x[7]   -2x[4]       =  0  C
```

In the lists the letter C denotes the contractible primitive relations of the varieties T and W respectively.

We try to compute all primitive relations of $W = \mathrm{Bl}_V(T)$ working as the algorithm `CBlowup`.

Divide primitive collections of T in the sets A, B, C:

set A	P_1, P_2, P_5
set B	/
set C	P_3, P_4, P_6, P_7

Consider the set A: its elements are primitive collections for the fan Σ_W and observe that the primitive relations associated to P_1 and P_5 do not change after the blow-up. Instead, the primitive relation associated to P_2 changes because in the negative part we find all generators of the cone of the variety V. Then $r^Y(P_2)$ is:

$$r^Y(P_2): \quad \texttt{x[1] + x[5] - x[8]=0.}$$

We compute the elements of set C':

$$\begin{aligned} P'_3 &= \texttt{\{x[4], x[8]\},} \\ P'_4 &= \texttt{\{x[6], x[8]\},} \\ P'_6 &= \texttt{\{x[3], x[8]\},} \\ P'_7 &= \texttt{\{x[5], x[6], x[8]\}.} \end{aligned}$$

The set P'_7 is not minimal because it contains P'_4, so the primitive collections of type c') are: P'_3, P'_4, P'_6.

Here `CBlowup` is not able to compute the relations associated to P'_6 and P'_4.

Let's consider P'_6:

$$
\begin{aligned}
\texttt{x[3] + x[8]} &= \texttt{x[3] + x[2] + x[7]}\\
&= \texttt{x[1] + x[2]+ x[6]}\\
&= \texttt{x[1] + x[3]+ x[5]}\\
&= \texttt{x[2] + x[3] +x[7]}\\
&= \quad \dots
\end{aligned} \tag{5.2}
$$

because `x[2]` + `x[7]` − `x[8]` = `0`, then in the negative part we find all generators of primitive relation `{x[2], x[6]}`, so we can use the primitive relation `x[2]` + `x[6]` − `x[3]` + `x[5]` = `0`. In this way we obtain a negative part which contains the primitive relation `{x[1], x[5]}`. Using its primitive relation we observe that the first row is equal to the last one so we do not continue.

Let's consider P_4':

$$
\begin{aligned}
\texttt{x[6] + x[8]} &= \texttt{x[6] + x[2] + x[7]}\\
&= \texttt{x[3] + x[5] + x[7]}\\
&= \texttt{x[1] + x[5] + x[6]}\\
&= \texttt{x[2] + x[6] + x[7]}\\
&= \quad \dots
\end{aligned} \tag{5.3}
$$

because `x[2]` + `x[7]` − `x[8]` = `0` and `x[2]` + `x[6]` − `x[3]` − `x[5]` = `0`. The set of generators in the negative part contains `{x[3], x[7]}`. Its primitive relation, `x[3]` + `x[7]` − `x[1]` − `x[6]` = `0`, determines a new negative part. Its set of variables contains the primitive collection `{x[1], x[5]}`, so when we sum the associated primitive relation we obtain a situation as in (5.2): again, we find that the first row is equal to the last so we do not continue.

The cases (5.2) and (5.3) create an infinity cycle in the algorithm `CBlowup`.

5.3 The new algorithm: $\mathrm{Blowup}[\mathbf{X}, \mathbf{cone}]$

In this section, we present the algorithm computing the blow-up $Y = \mathrm{Bl}_V(X)$.

Algorithm 5.3.1. Blowup[X, cone].

1. The algorithm tests if X is described by the primitive relations or not.

   (a) If X is described by the data of generators and maximal cones then the algorithm uses Definition 5.1.1 to computes the list describing Y. Its answer is the list

   $$\{n, t+1, \mathrm{Null}, \mathrm{G}^Y, \mathrm{MC}^Y\}.$$

   (b) If X is described by primitive relations then the algorithm proceeds in the following way:

      i. it computes all primitive collections of Y using Corollary 5.1.4;

      ii. then it computes the primitive relations associated to primitive collections of types a), b), c), using Lemmas 5.1.5, 5.1.6 and 5.1.7;

iii. it computes the primitive relations of type c') solving system of linear Diophantine equations;

iv. it gives the list which describes Y. The list has one among the following forms:

$$\{n, t+1, \mathrm{PR}^Y, \mathrm{Null}, \mathrm{Null}\};$$
$$\{n, t+1, \mathrm{PR}^Y, \mathrm{G}^Y, \mathrm{Null}\};$$
$$\{n, t+1, \mathrm{PR}^Y, \mathrm{G}^Y, \mathrm{MC}^Y\}.$$

If X is given by its fan Σ_X we use Definition 5.1.1 and describe Y with Σ_Y.

If X is given by its primitive relations, we proceed as in `CBlowup` to compute $PC(Y)$ and all primitive relations except those of type c'). In order to compute primitive relation of type c'), we use the algorithm `PrimRel` (see Chapter 2 and Section A.13.3).

This algorithm is not very fast, because it requires of solving a linear system for every primitive relation of type c').

Chapter 6

Toric blow-downs

In this chapter we will present the algorithm to compute the blow-down of a smooth complete toric variety Y with respect to a contractible primitive relation of form $x_1 + \cdots + x_r - x = 0$. The new variety is again a smooth, complete, toric variety.

Corollary 4.9 in [Sa] is a complete reference to enumerate all primitive collections of Y. Using the properties of a primitive collection we have written an algorithm which says how primitive relations change after a blow-down.

This chapter is divided in following sections: in Section 6.1 we expose the construction of a blow-down analyzing the fans of varieties (see [E2]), then we explain the known results and the theory to write our algorithm. In Section 6.2 we describe the algorithm.

6.1 The blow-down of toric variety from the point of view of fans and primitive relations

Let Y be a smooth and complete toric variety of dimension n. Suppose that its fan Σ_Y has a contractible primitive relation of type $x_1 + \cdots + x_r - x = 0$.

Let $\psi : Y \to X$ be the contraction of this class. Then X is smooth and ψ is the blow-up of X along an invariant subvariety $V = V(\sigma)$, with $\sigma = \langle x_1, \ldots, x_r \rangle$, namely $Y = \mathrm{Bl}_V X$.

Here we are in the same situation of the previous chapter, but the point of view is different: given Y, we want an algorithm to describe X.

Definition 5.1.1 describes the relation among Σ_X and Σ_Y.

For what concerns primitive collections, we have the following:

Theorem 6.1.1 ([Sa], Corollary 4.9)**.** *The primitive collections of X are given by:*

1. $P \in PC(Y)$ *such that* $P \neq \{x_1, \ldots, x_r\}$ *and* $x \notin P$*;*
2. $(P \setminus \{x\}) \cup \{x_1, \ldots, x_r\}$*, where* $P \in PC(Y)$ *verifies that* $x \in P$ *and* $(P \setminus \{x\}) \cup S \notin PC(Y)$ *for any subset* $S \subseteq \{x_1, \ldots, x_r\}$*.*

To find all primitive collections of X we can proceed in the following way. We define the following sets:

$$\begin{aligned} A_1 &= \{P \in PC(Y) \mid x \in P\}, \\ A_2 &= \{P \in PC(Y) \mid x \notin P \text{ and } P \neq \{x_1, \dots, x_r\}\}, \\ A_3 &= \{(P \setminus \{x\}) \cup \{x_1, \dots, x_r\} \mid \forall P \in A_1\}, \\ A_4 &= \{P \in A_2 \mid P \cap \{x_1, \dots, x_r\} \neq \emptyset\} = \{R_1, \dots, R_m\}, \end{aligned}$$

and the set

$$A_5 = \{Q \in A_3 \mid R_k \nsubseteq Q \,\forall\; k = 1, \dots, m\}.$$

We can prove the following:

Proposition 6.1.2. *The set of all primitive collections of X is equal to the union of the sets A_2 and A_5.*

Proof. We have to prove that $PC(X) = A_2 \cup A_5$. First we prove that $A_2 \cup A_5 \subseteq PC(X)$ and then the other inclusion.

We prove that every element in $A_2 \cup A_5$ verifies one of the two conditions of Corollary 6.1.1.

Let $P \in A_2$, then $P \neq \{x_1, \dots, x_r\}$ and $x \notin P$, so P satisfies the first condition of Corollary 6.1.1. Let $Q \in A_5$. In particular, $Q \in A_3$, so $Q = (P \setminus \{x\}) \cup \{x_1, \dots, x_r\}$ for some $P \in PC(Y)$ such that $x \in P$. Suppose by contradiction that $\widetilde{P} = (P \setminus \{x\}) \cup S \in PC(Y)$ for some $S \subseteq \{x_1, \dots, x_r\}$. Since $P \setminus \{x\} \neq \emptyset$ we have $\widetilde{P} \neq S$, hence $\widetilde{P} \in A_2$. Moreover by definition of primitive collection, $P \setminus \{x\}$ generates a cone in Σ_Y, hence $S \neq \emptyset$. This implies that $\widetilde{P} \cap \{x_1, \dots, x_r\} \neq \emptyset$, namely $\widetilde{P} \in A_4$. But $\widetilde{P} \subseteq Q$ and this contradicts the assumption $Q \in A_5$. So Q satisfies the second condition of Corollary 6.1.1.

Viceversa we consider $P \in PC(X)$. It satisfies either the first or the second condition of Corollary 6.1.1.

Let's suppose that it verifies the first one: then $P \in A_2$.

Let $Q \in PC(X)$ be such that $Q = (P \setminus \{x\}) \cup \{x_1, \dots, x_r\}$ for some $P \in A_1$. Then $Q \in A_3$ and Q strictly contains $\{x_1, \dots, x_r\} \in PC(Y)$, hence $Q \notin PC(Y)$. By what we proved above, we have that $A_4 \subseteq A_2 \subseteq PC(X)$. This implies that Q can not contain any element of A_4, so $Q \in A_5$.

□

To describe how primitive relation changes we denote by σ_P^X, σ_P^Y the cone associated to the primitive collection when P is a primitive collection for X and Y respectively. We use the same notation for primitive relations: $r^X(P)$ and $r^Y(P)$ is primitive relation associated to P when P is a primitive collection for X and Y respectively.

Lemma 6.1.3. *Let $P \in A_2$ be. Then P is a primitive collection for X.*

If $x \notin \sigma_P^Y$ then $r^X(P) = r^Y(P)$.

If $x \in \sigma_P^Y$, let $r^Y(P)$ be:

$$z_1 + \cdots + z_h - (ax + b_1y_1 + \cdots + b_sy_s) = 0$$

with $t, a, b_i \in \mathbb{Z}_{>0}$ and $s \in \mathbb{Z}_{\geq 0}$. Then $P \cap \{x_1, \ldots, x_r\} = \emptyset$ and

$$z_1 + \cdots + z_h - (ax_1 + \cdots + ax_r + b_1y_1 + \cdots + b_sy_s) = 0$$

is $r^X(P)$.

Proof. If $x \notin \sigma_P^Y$ then σ_P^Y is a cone in Σ_X then $\sigma_P^X = \sigma_P^Y$ and $r(P)^X = r(P)^Y$.

Suppose that $x \in \sigma_P^Y$ then $P \cap \{x_1, \ldots, x_r\} = \emptyset$. In fact, by contradiction, if $P \cap \{x_1, \ldots, x_r\} \neq \emptyset$ then P is a primitive collection of type c) for Y and $\sigma_P^X = \sigma_P^Y$. It is impossible, hence the intersection $P \cap \{x_1, \ldots, x_r\}$ is empty and $\sigma_P^X \neq \sigma_P^Y$. If $r^Y(P)$ is:

$$z_1 + \cdots + z_h - (ax + b_1y_1 + \cdots + b_sy_s) = 0$$

then $r^X(P)$ is obtained replacing x with the sum $x_1 + \cdots + x_r$ in the negative part of $r^Y(P)$:

$$z_1 + \cdots + z_h - (ax_1 + \cdots + ax_r + b_1y_1 + \cdots + b_sy_s) = 0.$$

□

Lemma 6.1.4. *Let $P \in A_5$, $P = (Q \setminus \{x\}) \cup \{x_1, \ldots, x_r\}$ with $Q \in A_1$.*

Then $\sigma_P^X = \sigma_Q^Y$, $G(\sigma_P^X) \cap \{x_1, \ldots, x_r\} = \emptyset$ and $r^X(P)$ is obtained by replacing x with the sum $x_1 + \cdots + x_r$ in the positive part of $r^Y(Q)$.

Proof. Since P strictly contains $\{x_1, \ldots, x_r\}$ and by Proposition 1.3.5 then $G(\sigma_P^X) \cap \{x_1, \ldots, x_r\} = \emptyset$. By definition, Q is a primitive collection of type b) then $\sigma_P^X = \sigma_Q^Y$. Moreover $r^X(P)$ is obtained by replacing x with the sum $x_1 + \cdots + x_r$ in the positive part of $r^Y(Q)$, by Lemma 5.1.6. □

6.2 Algorithm constructing blow-down

In this section we present the algorithm to compute the blow-down. It works with both representations of variety Y but it can verify that $x_1 + \cdots + x_k - x = 0$ is contractible only if the variety is described by the list of primitive relations.

Algorithm 6.2.1. Blowdown[Y, x].

1. The algorithm tests if Y is described by maximal cones or not.

    (a) If Y is described by the data of generators and maximal cones then the algorithm uses Definition 5.1.1 to compute the list describing X. Its answer is the list

    $$\{n, t-1, \text{Null}, \text{G}^X, \text{MC}^X\}.$$

(b) If Y is described by primitive relations then the algorithm proceeds in the following way:

i. it computes the sets $A_1, \ldots, A_5$ and, then, $PC(X)$ using Proposition 6.1.2;

ii. it computes all primitive relations using Lemmas 6.1.3 and 6.1.4;

iii. the output of the algorithm is one of the following lists:

$$\{n, t-1, \mathrm{PR}^X, \mathrm{Null}, \mathrm{Null}\};$$
$$\{n, t-1, \mathrm{PR}^X, \mathrm{G}^X, \mathrm{Null}\};$$
$$\{n, t-1, \mathrm{PR}^X, \mathrm{G}^X, \mathrm{MC}^X\}.$$

Notice that in this case we does not need to use the algorithm `PrimRel` to compute the primitive relations.

See Section A.7 and next chapter for examples.

Chapter 7

Examples and applications

In this section we present five interesting examples. We describe them using the algorithms of the package *Toric Varieties*. Moreover we analyze the information which we obtain from the answers of the algorithms.

7.1 The Oda Threefold

The Oda threefold is a complete, non projective toric threefold with Picard number $\rho = 4$ described on page 84 in [O2]. Since all complete toric manifolds with $\rho \leq 3$ are projective (see [KlSt]), the Oda threefold is one of the first examples of non-projective toric manifolds.

The Oda threefold is obtained from $\mathbb{P}^3$ by four blow-ups and one blow-down. Let's construct it using the programs of the package.

Let Z be the projective space of dimension 3 and consider the following blow-ups:

```
In[1]:= Z = ProjSpace[3]

In[2]:= Z1 = Blowup[Z, {x[2], x[4]}];

In[3]:= Z2 = Blowup[Z1, {x[3], x[4]}];

In[4]:= Z3 = Blowup[Z2, {x[1], x[4]}];

In[5]:= W = Blowup[Z3, {x[1], x[5]}];
```

W is a projective threefold with $\rho_W = 5$. Let's see its primitive relations:

```
In[6]:= TabVar[W]
```

x[1] + x[4]	- x[7]	=	0		
x[1] + x[5]	- x[8]	=	0	C	γ_1
x[2] + x[4]	- x[5]	=	0		
x[2] + x[6]	- x[3] - x[5]	=	0	C	γ_2
x[2] + x[7]	- x[8]	=	0	C	γ_3
x[3] + x[4]	- x[6]	=	0		
x[3] + x[7]	- x[1] - x[6]	=	0	C	γ_4
x[3] + x[8]		=	0		
x[4] + x[8]	- x[5] - x[7]	=	0	C	γ_5
x[6] + x[8]	- x[4]	=	0		
x[5] + x[6] + x[7]	- 2x[4]	=	0	C	γ_6

W has six contractible primitive relations: $\gamma_1, \dots, \gamma_6$.The associated contractions are all birational: φ_{γ_1} and φ_{γ_3} are smooth blow-ups with the same exceptional divisor $V(x_8)$; φ_{γ_6} contracts a divisor to a singular point, φ_{γ_2}, φ_{γ_4} and φ_{γ_5} are small contractions.

Let's check which contractible classes are extremal:

```
In[7]:= TabExtremalCurves[W]

Out[7]:=
```

x[2] - x[3] - x[5] + x[6]	$->$	-1 1 -1 0 0	E	γ_2
x[1] + x[5] - x[8]	$->$	1 0 0 0 0	E	γ_1
-x[1] + x[3] - x[6] + x[7]	$->$	0 0 1 0 0	E	γ_4
x[4] - x[5] - x[7] + x[8]	$->$	0 0 0 1 0	E	γ_5
-2 x[4] + x[5] + x[6] + x[7]	$->$	0 0 0 0 1	E	γ_6
x[2] + x[7] - x[8]	$->$	0 1 0 0 0		γ_3

We see that γ_3 is non-extremal: hence $\mathrm{NE}(W)$ is a five dimensional simplicial cone generated by γ_1, γ_2, γ_4, γ_5, γ_6. We also see that $\gamma_3 = \gamma_1 + \gamma_2 + \gamma_4$. The equivariant morphism $\varphi_{\gamma_3} : W \to T$ is a smooth blow-down. T is a smooth, complete toric variety, but by Proposition 1.5.2, not projective. This is the Oda threefold:

```
In[8]:= T = Blowdown[W, x[8]];
```

The primitive relations of T are:

```
In[9]:= TabVar[T]

Out[9]:=
```

x[1] + x[4]	- x[7]	=	0		
x[1] + x[5]	- x[2] - x[7]	=	0	C	η_1
x[2] + x[4]	- x[5]	=	0		
x[2] + x[6]	- x[3] - x[5]	=	0	C	η_2
x[3] + x[4]	- x[6]	=	0		
x[3] + x[7]	- x[1] - x[6]	=	0	C	η_4
x[5] + x[6] + x[7]	-2 x[4]	=	0	C	η_6

We can see at once that T is not projective because there is no primitive relation with zero negative part (see Proposition 4.3.4). In fact:

```
In[10]:= ProjQ[T]

Out[10]:= False
```

T has four contractible classes η_1, η_2, η_4 and η_6 which are images of the corresponding γ_j under the push-forward $\varphi_* : \mathcal{N}_1(W) \to \mathcal{N}_1(T)$. The morphism associated to η_6 is a birational contraction sending a divisor to a singular point. For $i = 1, 2, 4$, η_i is the numerical class of an invariant curve $C_i \cong \mathbb{P}^1$ with normal bundle $\mathcal{N}_{C_i/T} = \mathcal{O}_{\mathbb{P}^1}(-1)^{\oplus 2}$. Observe that C_1 is the center of the blow-up $W \to T$. We have that $\eta_1 + \eta_2 + \eta_4 = 0$, namely $C_1 + C_2 + C_4$ is numerically equivalent to zero in T. The Mori cone $\mathrm{NE}(T)$ has dimension 4 and contains the plane spanned by η_1, η_2. Notice that in this case, contractible classes do not generate all $\mathcal{N}_1(T)$.

7.2 Threefolds with no non-trivial nef line bundles

In [FP], Fujino and Payne analyze some examples of smooth, complete toric threefolds with no non-trivial nef line bundles. If X is such a threefold, we have *Nef*$(X) = \{0\}$ and $\mathrm{NE}(X) = \mathcal{N}_1(X)_{\mathbb{Q}}$, namely every 1-cycle in X is numerically equivalent to an effective curve. Such X is not projective, and it admits no non-constant morphisms to projective varieties. In [FP], it is shown that $\rho_X \geq 5$ and all cases with $\rho_X = 5$ are described. They consist of two infinite families plus one exceptional example: X_1.

We are going to describe X_1 and the two families.

Since X_1 is obtained by desingularization of a singular variety Y we begin to describe the fan of variety Y and then its desingularization.

Let Σ be the fan in $\mathbb{Q}^3$ whose rays are generated by

```
x[1] = (1,0,0),    x[2] = (0,1,0),    x[3] = (0,0,1),
x[4] = (0,-1,-1),  x[5] = (-1,0,-1),  x[6] = (-2,-1,0),
```

and whose maximal cones are:

⟨x[1], x[2], x[3]⟩, ⟨x[1], x[2], x[4]⟩, ⟨x[2], x[4], x[5]⟩,
⟨x[2], x[3], x[5]⟩, ⟨x[3], x[5], x[6]⟩, ⟨x[1], x[3], x[6]⟩,
⟨x[1], x[4], x[6]⟩, ⟨x[4], x[5], x[6]⟩.

The fan Σ is not regular, in fact the last cone ⟨x[4], x[5], x[6]⟩ is not generated by a part of a basis of the lattice $N = \mathbb{Z}^3$. The invariant point P corresponding to the cone ⟨x[4], x[5], x[6]⟩ is the unique singular point of Y, the variety associated to Σ. We replace it by the following three cones:

⟨x[4], x[5], x[7]⟩, ⟨x[4], x[6], x[7]⟩, ⟨x[5], x[6], x[7]⟩,

where x[7] = 2/3 x[4] + 1/3 x[5] + 1/3 x[6].

We obtain a new fan Σ_1 which is not yet regular, because the cone $\langle$`x[5], x[6], x[7]`$\rangle$ is not generated by a basis of N. Let Y_1 the variety associated to Σ_1. Again, we define a new generator `x[8] = 1/2 (x[5] + x[6] + x[7])` and replace the cone $\langle$`x[5], x[6], x[7]`$\rangle$ with the cones:

$\langle$`x[5], x[6], x[8]`$\rangle$, $\langle$`x[6], x[7], x[8]`$\rangle$, $\langle$`x[5], x[7], x[8]`$\rangle$.

Then we obtain a new variety whose fan, denoted by Δ, is generated by

```
x[1] = (1,0,0),     x[2] = (0,1,0),     x[3] = (0,0,1),
x[4] = (0,-1,-1),   x[5] = (-1,0,-1),   x[6] = (-2,-1,0),
x[7] = (-1,-1,-1),  x[8] = (-2,-1,-1).
```

Its maximal cones are:

$\langle$`x[1], x[2], x[3]`$\rangle$, $\langle$`x[1], x[2], x[4]`$\rangle$, $\langle$`x[2], x[4], x[5]`$\rangle$,
$\langle$`x[2], x[3], x[5]`$\rangle$, $\langle$`x[3], x[5], x[6]`$\rangle$, $\langle$`x[1], x[3], x[6]`$\rangle$,
$\langle$`x[1], x[4], x[6]`$\rangle$, $\langle$`x[4], x[5], x[7]`$\rangle$, $\langle$`x[4], x[6], x[7]`$\rangle$,
$\langle$`x[5], x[6], x[8]`$\rangle$, $\langle$`x[5], x[7], x[8]`$\rangle$, $\langle$`x[6], x[7], x[8]`$\rangle$.

We call X_1 the variety associated to the fan Δ. It is a smooth, complete, toric variety, hence it is a resolution of Y:

$$X_1 \longrightarrow Y_1 \longrightarrow Y.$$

If we compute the primitive relations of X_1, then we can verify that X_1 is not projective because there is no primitive relation with zero negative part:

```
In[1]:= TabVar[X1]

Out[1]:=
```

`x[1] + x[5]`	`- x[2] - x[4]`	=	0	C	γ_1
`x[1] + x[7]`	`- x[4]`	=	0		
`x[1] + x[8]`	`- x[7]`	=	0		
`x[2] + x[6]`	`-2 x[3] -2 x[5]`	=	0	C	γ_2
`x[2] + x[7]`	`- x[5]`	=	0		
`x[2] + x[8]`	`- x[3] -2 x[5]`	=	0		
`x[3] + x[4]`	`-2 x[1] - x[6]`	=	0	C	γ_3
`x[3] + x[7]`	`- x[1] - x[6]`	=	0		
`x[3] + x[8]`	`- x[6]`	=	0		
`x[4] + x[8]`	`-2 x[7]`	=	0	C	γ_4
`x[4] + x[5] + x[6]`	`- x[7] - x[8]`	=	0		
`x[5] + x[6] + x[7]`	`-2 x[8]`	=	0	C	γ_5

Let's consider the set of all numerical classes of invariant curves in X_1 and compute their coordinates with respect to a basis B of $\mathcal{N}_1(X)$ contained in $\mathcal{I}$:

```
In[2]:= CoordCurveClasses[X1]

Out[2]:=
```

x[2] + x[3] + x[4]	$\longrightarrow$	1 0 1 1 1	
2x[1] + x[2] + x[6]	$\longrightarrow$	1 2 0 2 1	
-2x[1] + x[3] + x[4] - x[6]	$\longrightarrow$	0 -2 1 -1 0	γ_3
x[1] + x[3] + x[5]	$\longrightarrow$	0 1 1 1 1	
x[1] - x[2] - x[4] + x[5]	$\longrightarrow$	-1 1 0 0 0	γ_1
x[2] - 2x[3] - 2x[5] + x[6]	$\longrightarrow$	1 0 -2 0 -1	γ_2
x[2] - x[5] + x[7]	$\longrightarrow$	1 0 0 0 0	
x[1] - x[4] + x[7]	$\longrightarrow$	0 1 0 0 0	
2x[4] + x[5] + x[6] - 3x[7]	$\longrightarrow$	0 0 0 2 1	
x[3] - x[6] + x[8]	$\longrightarrow$	0 0 1 0 0	
x[4] - 2x[7] + x[8]	$\longrightarrow$	0 0 0 1 0	γ_4
x[5] + x[6] + x[7] - 2x[8]	$\longrightarrow$	0 0 0 0 1	γ_5

We can observe that in the basis above $3\gamma_1 + 2\gamma_2 + 2\gamma_3$ has coordinates $(-1,-1,-2,-2,-2)$. The only convex cone in $\mathbb{Q}^5$ containing the canonical basis and the point $(-1,-1,-2,-2,-2)$ is the whole $\mathbb{Q}^5$, so we see that $Nef(X_1) = \{0\}$ and $\mathrm{NE}(X_1) = \mathcal{N}(X_1)_{\mathbb{Q}}$.

Notice that X_1 has five contractible classes: $\gamma_1, \ldots, \gamma_5$. The morphism associated to γ_5 is a birational contraction sending a divisor to a singular point, instead one associated to γ_4 is a birational contraction sending a divisor to a singular curve. The other are small contractions. For every $i = 1, \ldots, 5$, let C_i be an invariant curve with numerical class γ_i. We can compute their normal bundle. We have that:

$$\begin{aligned}
\mathcal{N}_{C_1/X_1} &= \mathcal{O}_{\mathbb{P}^1}(-1)^{\oplus 2},\\
\mathcal{N}_{C_2/X_1} &= \mathcal{O}_{\mathbb{P}^1}(-2)^{\oplus 2},\\
\mathcal{N}_{C_3/X_1} &= \mathcal{O}_{\mathbb{P}^1}(-2) \oplus \mathcal{O}_{\mathbb{P}^1}(-1),\\
\mathcal{N}_{C_4/X_1} &= \mathcal{O}_{\mathbb{P}^1}(-2) \oplus \mathcal{O}_{\mathbb{P}^1},\\
\mathcal{N}_{C_5/X_1} &= \mathcal{O}_{\mathbb{P}^1}(-2) \oplus \mathcal{O}_{\mathbb{P}^1}(1).
\end{aligned}$$

The variety X_1 is labelled as [8-12] in [O1] on page 79, where the authors fix as basis of the lattice the vectors $n = (-1,-1,-1)$, $n' = (1,0,0)$, $n'' = (0,0,1)$.

Let's consider now the two families constructed in [FP] Example 2 and Example 3. The first family has the following generators:

```
x[1] = (1,0,0),   x[2] = (0,1,0),    x[3] = (0,0,1),    x[4] = (0,-1,-a),
x[5] = (0,0,-1),  x[6] = (-1,1,-1),  x[7] = (-1,0,-1),  x[8] = (-1,-1,0),
```

with $a \neq 0, -1$; and the maximal cones are:

⟨x[1], x[2], x[3]⟩, ⟨x[1], x[3], x[4]⟩, ⟨x[1], x[4], x[5]⟩,
⟨x[1], x[5], x[6]⟩, ⟨x[1], x[2], x[6]⟩, ⟨x[2], x[3], x[8]⟩
⟨x[3], x[4], x[8]⟩, ⟨x[4], x[5], x[8]⟩, ⟨x[5], x[6], x[7]⟩,
⟨x[5], x[7], x[8]⟩, ⟨x[6], x[7], x[8]⟩, ⟨x[2], x[6], x[8]⟩.

This family is labelled [8-5′] in [O1] on page 78, we can prove it choosing $n = (1, 0, 0)$, $n' = (0, 1, 0)$ and $n'' = (0, 0, 1)$ as basis of the lattice.

Fix $a = 1$, then we obtain a smooth complete toric variety, called X_2. X_2 is described by the following primitive relations:

```
In[3]:= TabVar[X2]

Out[3]:=

          x[1] + x[7]          - x[7]            =  0
          x[1] + x[8]          - x[3] - x[4]     =  0  C
          x[2] + x[4]          - x[5]            =  0
          x[2] + x[5]          - x[1] - x[6]     =  0  C
          x[2] + x[7]          - x[6]            =  0
          x[3] + x[5]                            =  0
          x[3] + x[6]          -2 x[2] - x[8]    =  0  C
          x[3] + x[7]          - x[2] - x[8]     =  0
          x[4] + x[6]          - x[5] - x[7]     =  0
          x[4] + x[7]          -2 x[5] - x[8]    =  0  C
          x[5] + x[6] + x[8]   -2 x[7]           =  0  C
```

In this case, we can observe that there is a primitive relation with negative part equal to zero, so we can not conclude at once that X_2 is not projective.

Let's consider the set $\mathcal{I}$ of all numerical classes of invariant curves in X_2, and compute their coordinates with respect to a basis B contained in $\mathcal{I}$:

```
In[4]:= CoordCurvesClasses[X2]

Out[4]:=

          x[1] + x[3] + x[6] - x[2]     →  1 0 0 0 0
          x[2] + x[4] + x[3]            →  1 -1 1 0 2
          x[3] + x[5]                   →  1 -1 0 -1 1
          x[1] + x[4] + x[6] - 2x[5]    →  0 1 1 2 0
          x[2] + x[5] - x[1] - x[6]     →  0 -1 0 -1 1
          x[1] + x[2] + x[8]            →  0 1 1 1 1
          x[3] + x[6] - 2x[2] - x[8]    →  1 -1 -1 -1 -1
          x[1] + x[8] - x[3] - x[4]     →  -1 2 0 1 -1
          x[1] + x[5] + x[8] - x[4]     →  0 1 0 0 0
          x[1] + x[7] - x[5]            →  0 1 0 1 0
          x[5] + x[6] + x[8] - 2x[7]    →  0 0 1 0 0
          x[4] + x[7] - 2x[5] - x[8]    →  0 0 0 1 0
          x[2] + x[7] - x[6]            →  0 0 0 0 1
```

Analyzing the coordinates we can see that:

$$\underline{w}_5 + \underline{w}_7 + \underline{w}_8 + \underline{w}_{11} + \underline{w}_{12} + \underline{w}_{13} = 0$$

where $\underline{w}_i$ denotes the i-th vector in the list above. Moreover, any five of the six vectors above are linearly independent, hence $\mathrm{NE}(X_2) = \mathcal{N}_1(X_2)_{\mathbb{Q}}$. The algorithm `ProjQ` confirms that X_2 is not projective:

```
In[5]:= ProjQ[X2]

Out[5]:= False
```

The second family constructed in [FP] has the following generators:

```
x[1] = (-1,b,0), x[2] = (0,-1,0), x[3] = (1,-1,0), x[4] = (-1,0,-1),
x[5] = (0,0,-1), x[6] = (0,1,0),  x[7] = (0,0,1),  x[8] = (1,0,a),
```

with $a, b \neq 0$ and $(a, b) \neq (\pm 1, \pm 1)$. The maximal cones of the fan are:

⟨x[1], x[2], x[3]⟩, ⟨x[1], x[3], x[4]⟩, ⟨x[1], x[4], x[5]⟩,
⟨x[1], x[5], x[6]⟩, ⟨x[1], x[2], x[6]⟩, ⟨x[2], x[3], x[8]⟩,
⟨x[3], x[4], x[8]⟩, ⟨x[4], x[5], x[8]⟩, ⟨x[5], x[6], x[7]⟩,
⟨x[5], x[7], x[8]⟩, ⟨x[6], x[7], x[8]⟩, ⟨x[2], x[6], x[8]⟩.

The varieties of this family are labelled [8-13′] in [O1] on page 79, it is enough to chose $n = (1, -1, 0)$, $n' = (0, 1, 0)$, $n'' = (0, 0, 1)$ as basis of N.

7.3 Non projective threefolds that become projective after a blow-up

In this section we consider the last family of previous section because it is one of the simplest examples of families of smooth, complete, toric varieties with $\rho = 5$ which becomes projective after a blow-up. There is a detailed description of X in [O1] and in [Bo]. In [Bo], Bonavero studied the projectivity of this family introducing it in a different way with respect to what Fujino and Payne do in [FP].

Let Σ be the fan in $\mathbb{Q}^3$ generated by:

```
x[1] = (-1,d,0), x[2] = (0,-1,0), x[3] = (1,0,0), x[4] = (-1,-1,-1),
x[5] = (0,0,-1), x[6] = (0,1,0),  x[7]= (0,0,1),  x[8] = (1,1,c),
```

where c and d are two integer parameters. Its maximal cones are:

⟨x[1], x[2], x[3]⟩, ⟨x[1], x[3], x[4]⟩, ⟨x[1], x[4], x[5]⟩,
⟨x[1], x[5], x[6]⟩, ⟨x[1], x[2], x[6]⟩, ⟨x[2], x[3], x[8]⟩,
⟨x[3], x[4], x[8]⟩, ⟨x[4], x[5], x[8]⟩, ⟨x[5], x[6], x[7]⟩,
⟨x[5], x[7], x[8]⟩, ⟨x[6], x[7], x[8]⟩, ⟨x[2], x[6], x[8]⟩.

Let $X_{c,d}$ be the variety described by Σ.

Observe that we changed the names of two parameters because in this case we use Bonavero's notation: he sets $n = (1, 0, 0)$, instead of $n = (1, -1, 0)$ when he fixes the basis of N. From this choice it follows that the parameter b considered by Fujino and Payne in [FP] is equal to $d + 1$, while the other parameter, c, coincides with a.

Bonavero proves that X is non-projective if and only if $c \neq 0$ and $d \neq -1$ (see [Bo], Proposition 3).

Let's consider the variety X obtained for $c = 1$ and $d = 0$ (respectively $a = 1$ and $b = 1$). The fan of X is generated by:

```
x[1] = (-1,0,0),  x[2] = (0,-1,0),  x[3] = (1,0,0),  x[4] = (-1,-1,-1),
x[5] = (0,0,-1),  x[6] = (0,1,0),   x[7] = (0,0,1),  x[8] = (1,1,1).
```

Let's compute its primitive relations, hence X is described by:

```
In[1]:= TabVar[X]

Out[1]:=
```

x[1] + x[3]			=	0		
x[1] + x[5]	− x[4]	− x[6]	=	0	C	γ_1
x[1] + x[8]	− x[6]	− x[7]	=	0	C	γ_2
x[2] + x[5]	− x[3]	− x[4]	=	0	C	γ_3
x[2] + x[6]			=	0		
x[2] + x[8]	− x[3]	− x[7]	=	0	C	γ_4
x[3] + x[6]	− x[5]	− x[8]	=	0	C	γ_5
x[4] + x[7]	− x[1]	− x[2]	=	0	C	γ_6
x[4] + x[8]			=	0		
x[5] + x[7]			=	0		

In this case there are six contractible classes: $\gamma_1, \ldots, \gamma_6$ and their associated morphisms φ_{γ_i} are all small contractions.

Let's consider the set $\mathcal{I}$ of all numerical classes of invariant curves in X, and compute their coordinates with respect to a basis B contained in $\mathcal{I}$:

```
In[2]:= CoordCurvesClasses[X]

Out[2]:=
```

```
x[1] + x[3]                      -> -1 1 0 0 0
x[2] - x[3] - x[4] + x[5]        -> -1 0 -1 0 -1
x[1] - x[4] + x[5] - x[6]        -> -1 0 -1 -1 0
x[2] + x[6]                      -> -1 1 0 1 -1
x[3] + x[4] - x[5] + x[6]        -> 0 1 1 1 0
x[1] + x[2] + x[3] - x[7]        -> 1 0 0 0 0
- x[2] + x[3] + x[4] - x[7]      -> 0 1 0 0 0
x[5] + x[7]                      -> -1 1 -1 0 -1
- x[1] + x[4] + x[6] + x[7]      -> 0 1 0 1 -1
x[3] - x[5] + x[6] - x[8]        -> 0 0 1 0 0
x[3] + x[6] + x[7] - x[8]        -> -1 1 0 0 -1
x[4] + x[8]                      -> 0 1 0 1 0
x[2] - x[3] - x[7] + x[8]        -> 0 0 0 1 0
x[1] - x[6] - x[7] + x[8]        -> 0 0 0 0 1
```

Observe that there are two classes of curves such that their coordinates are all non positive with respect to the fixed basis in $\mathcal{N}_1(X)_{\mathbb{Q}} \cong \mathbb{Q}^5$. This allows us to conclude that the variety is not projective. Also the command `ProjQ` confirms that fact:

```
In[3]:= ProjQ[X]

Out[3]:= False
```

In this case, the Mori cone of X does not coincide with the whole space $\mathcal{N}_1(X)_{\mathbb{Q}}$, but it contains a linear subspace of $\mathcal{N}_1(X)_{\mathbb{Q}}$ of dimension 4.

There are other three varieties in the family $X_{c,d}$ with the same property, that is they are non projective varieties but their Mori cone does not coincide with $\mathbb{Q}^5$. They are obtained for the following value of parameters: $(c, d) = (1, -2), (-1, 0), (-1, -2)$.

Notice that Fujino and Payne exclude these four cases choosing $(a, b) \neq (\pm 1, \pm 1)$.

7.4 A 6-dimensional example

In this example, we present a smooth, complete, toric variety obtained considering some blow-ups of the projective space of dimension 6 along three invariant subvarieties.

Let S be the projective space of dimension 6. We consider the following blow-ups:

$$S_1 = \mathrm{Bl}_{C_1} S$$
$$S_2 = \mathrm{Bl}_{C_2} S_1$$
$$S_3 = \mathrm{Bl}_{C_3} S_2$$

where C_1, C_2, C_3 are three curves in the varieties S, S_1, S_2 respectively. C_1, C_2, C_3 respectively correspond to the cones $\langle$`x[1],x[2],x[3],x[4],x[5]`$\rangle$, $\langle$`x[1],x[2],x[3],x[4],x[6]`$\rangle$ and $\langle$`x[1],x[2],x[3],x[4],x[7]`$\rangle$.

```
In[1]:= S = ProjSpace[6];

In[2]:= S1 = Blowup[S, {x[1], x[2], x[3], x[4], x[5]}];

In[3]:= S2 = Blowup[S1, {x[1], x[2], x[3], x[4], x[6]}];

In[4]:= S3 = Blowup[S2, {x[1], x[2], x[3], x[4], x[7]}];
```

S_3 is a variety of dimension 6 and $\rho = 4$. It is described by the following primitive relations:

```
In[5]:= TabVar[S3]

Out[5]:=

      x[5] + x[9]  - x[6] - x[8]                            = 0  C
      x[5] + x[10]  - x[7] - x[8]                           = 0  C
      x[6] + x[10]  - x[7] - x[9]                           = 0  C
      x[6] + x[7] + x[8]                                    = 0
      x[7] + x[8] + x[9]  - x[1] - x[2] - x[3] - x[4]       = 0
      x[8] + x[9] + x[10]  -2 x[1] -2 x[2] -2 x[3] -2 x[4]  = 0  C
      x[1] + x[2] + x[3] + x[4] + x[5]  - x[8]              = 0
      x[1] + x[2] + x[3] + x[4] + x[6]  - x[9]              = 0
      x[1] + x[2] + x[3] + x[4] + x[7]  - x[10]             = 0  C
```

There are five contractible primitive relations, they are denoted by the letter `C` in the previous table. The associated morphisms are all birational: the morphisms associated to the first three classes have an exceptional locus of dimension 4 and when they are restricted to the exceptional locus, they are $\mathbb{P}^1$-bundles. The morphism associated to the fourth class contracts the surface $V(\langle$`x[1], x[2], x[3], x[4]`$\rangle)$ to a singular point. The last one is a smooth blow-up with exceptional divisor $V($`x[10]`$)$.

Since S_3 is obtained by a blowup of $\mathbb{P}^6$ then it is projective and we can compute which contractible classes are extremals:

```
In[6]:= TabExtremalClasses[S3]

Out[6]:=

 x[5] - x[6] - x[8] + x[9]                               -> 1 -1 0 0  E
 x[6] - x[7] - x[9] + x[10]                              -> 0 1 0 0   E
 -2 x[1] -2 x[2] -2 x[3] -2 x[4] + x[8] + x[9] + x[10]   -> 0 0 1 0   E
 x[1] + x[2] + x[3] + x[4] + x[7] - x[10]                -> 0 0 0 1   E
 x[5] - x[7] - x[8] + x[10]                              -> 1 0 0 0
```

There are four extremal classes and only one non-extremal.

7.5 An almost Fano threefold with $\rho = 35$

In this section we present a smooth, projective and almost Fano toric variety X of dimension 3 and $\rho = 35$.

A smooth, projective variety X is almost Fano if its anticanonical bundle $-K_X$ is nef and big. In the toric case, we know that for every dimension n, toric almost Fano varieties are in finite number (see [Ba3]).

In this case it is convenient to use the concept of polytope to introduce the variety. A polytope is the convex hull of finitely many points.

When we consider Gorestein toric Fano varieties of dimension n, we see that they correspond bijectively to some special polytopes: *reflexive polytopes* (the definition was introduced by Batyrev in [Ba3]). Reflexive polytopes have the property that their vertices are integral (i. e. in $\mathbb{Z}^n$), and the origin is their unique interior integral point. An almost Fano variety of dimension n is obtained by a crepant refinement of a reflexive polytope (see [Ba3], [N]).

In dimension 3, Kreuzer and Skarke give a complete classification of all reflexive polytopes (see [KrSk], [list]). They are 4319. After this classification, we know that any almost Fano threefold X has $\rho_X \leq 35$. We are now going to describe explicitly such X with maximal Picard number $\rho_X = 35$. In [list], we see that there are only two 3-dimensional reflexive polytopes containing 39 integral points; both are simplices. We consider one of them: the simplex $\mathcal{P} \subset \mathbb{Z}^3$ with vertices:

$$v_1 = (1, 0, 0), \quad v_2 = (1, 2, 0), \quad v_3 = (1, 2, 6), \quad v_4 = (-5, -4, -6).$$

In order to obtain our example, X we have to triangulate each facet of $\mathcal{P}$. The vertices of each triangle must be a basis of $\mathbb{Z}^3$. The fan of X will be given by the cones over all these triangles. We describe all 38 integral points on the facets of $\mathcal{P}$ and the triangulations that we have chosen.

On F_1, the facet of vertices v_1, v_2, v_3, we have the following integral points:

```
x[27] = (1,0,0), x[28] = (1,1,0), x[29] = (1,1,1),
x[30] = (1,1,2), x[31] = (1,1,3), x[32] = (1,2,0),
x[33] = (1,2,1), x[34] = (1,2,2), x[35] = (1,2,3),
x[36] = (1,2,4), x[37] = (1,2,5), x[38] = (1,2,6).
```

On F_2, the facet of vertices v_1, v_2, v_4 there are the following integral points:

```
x[1] = (-5,-4,-6), x[2] = (-4,-3,-5), x[4] = (-3,-2,-4),
x[7] = (-2,-2,-3), x[8] = (-2,-1,-3), x[12] = (-1,-1,-2),
x[14] = (-1,0,-2), x[19] = (0,0,-1),  x[21] = (0,1,-1),
x[27] = (1,0,0),   x[28] = (1,1,0),   x[32] = (1,2,0).
```

On F_3, the facet of vertices v_1, v_3, v_4, we have these integral points:

```
x[1] = (-5,-4,-6), x[3] = (-4,-3,-4), x[6] = (-3,-2,-2),
x[7] = (-2,-2,-3), x[11] = (-2,-1,0), x[13] = (-1,-1,-1),
x[18] = (-1,0,2),  x[20] = (0,0,1),   x[26] = (0,1,4),
x[27] = (1,0,0),   x[31] = (1,1,3),   x[38] = (1,2,6).
```

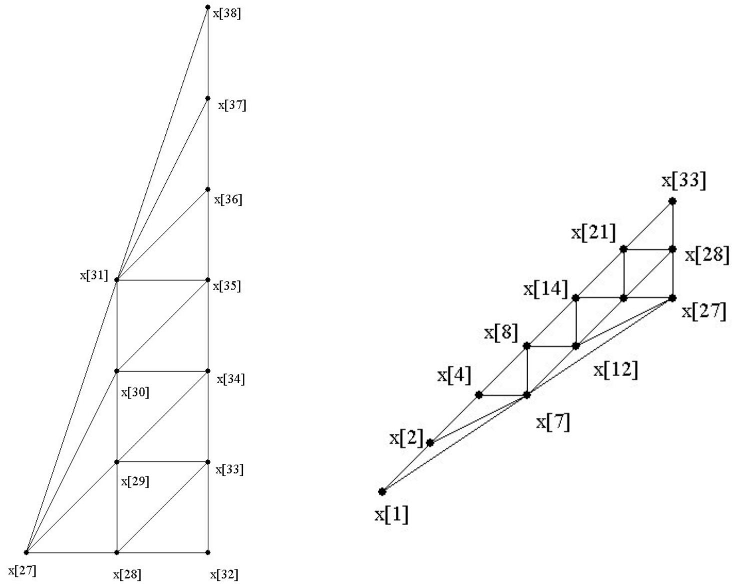


Figure 7.1: Triangulation of F_1 (on the left) and of F_2 (on the right).

On F_4, the facet of vertices v_2, v_3, v_4 we find the following integral points:

```
x[1] = (-5,-4,-6),  x[2] = (-4,-3,-5),  x[3] = (-4,-3,-4),
x[4] = (-3,-2,-4),  x[5] = (-3,-2,-3),  x[6] = (-3,-2, 2),
x[8] = (-2,-1,-3),  x[9] = (-2,-1,-2),  x[10] = (-2,-1,-1),
x[11] = (-2,-1,0),  x[14] = (-1,0,-2),  x[15] = (-1,0,-1),
x[16] = (-1,0,0),   x[17] = (-1,0,1),   x[18] = (-1,0,2),
x[21] = (0,1,-1),   x[22] = (0,1,0),    x[23] = (0,1,1),
x[24] = (0,1,2),    x[25] = (0,1,3),    x[26] = (0,1,4),
x[31] = (1,1,3),    x[32] = (1,2,0),    x[33] = (1,2,1),
x[34] = (1,2,2),    x[35] = (1,2,3),    x[36] = (1,2,4),
x[37] = (1,2,5),    x[38] = (1,2,6).
```

The triangulations described in Figures 7.1 and 7.2, define the fan of X. Observe that Σ_X has 38 generators, hence $\rho_X = 35$. Moreover $-K_X$ is nef and big.

Defined the fan, we have computed its primitive collections. They are 596 all of cardinality 2, except one of cardinality 3. Since the set of the generators is too large, our program `PrimRel` is not able to compute the associated primitive relations. In fact for every primitive collection the command has to solve a linear system with 36 (or 35) unknowns. Given a primitive collection Q, we have solved the problem

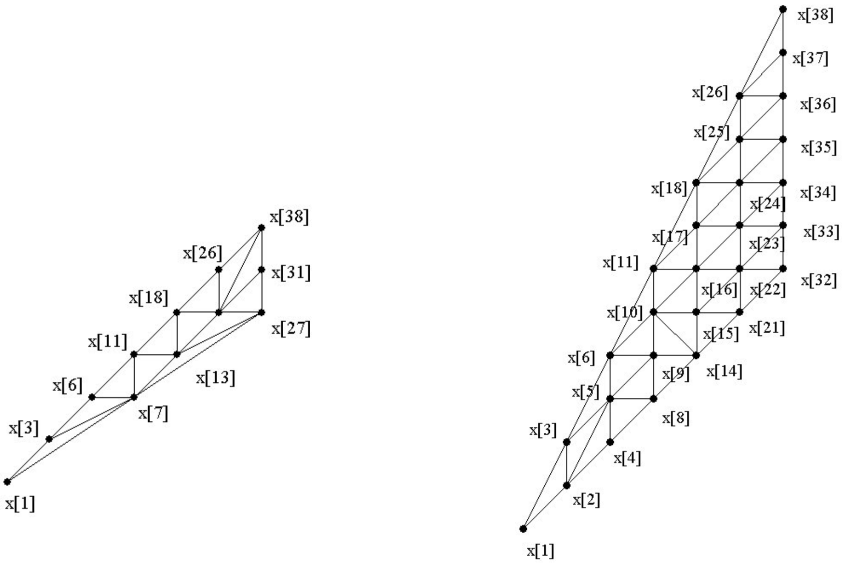


Figure 7.2: Triangulation of F_3 (on the left) and of F_4 (on the right).

considering for every maximal cone $T \in \Sigma_X(3)$ the following linear system:

$$\sum_{x \in Q} x = \sum_{y_i \in G(T)} a_i y_i$$

where y_i are the generators of T and a_i are the unknowns of the system.

We consider a maximal cone such that the linear system above has a solution in $\mathbb{Z}^3_{\geq 0}$. In this way we obtain the coefficients of the negative part of $r(Q)$.

Using the algorithm `ProjQ`, we can verify that X is projective:

```
In[1]:= ProjQ[X]

Out[1]:= True
```

Using the command `TabContrRel` we found all contractible classes of X. Since they are 56, we do not give the list, but we describe their properties. There are 51 classes of type:

```
x[i] + x[j] - x[h] -  x[k] = 0
```

where $i, j, k, h \in \{1, \ldots, 38\}$. The morphism associated to these classes are small contractions with exceptional locus a curve $C \cong \mathbb{P}^1$ with normal bundle $\mathcal{N}_{C/X} = \mathcal{O}_{\mathbb{P}^1}(-1)^{\oplus 2}$.

Then there are the following contractible classes:

$$\gamma_1: \quad \texttt{x[2] + x[8] -2 x[4] = 0}$$
$$\gamma_2: \quad \texttt{x[20] + x[37] - x[38] = 0}$$
$$\gamma_3: \quad \texttt{x[22] + x[28] - x[32] = 0}$$
$$\gamma_4: \quad \texttt{x[36] + x[38] -2 x[37] = 0}$$

The morphisms associated to the classes γ_1 and γ_4 are birational and send a divisor to a singular curve, while the morphisms associated to the classes γ_2, γ_3 are two smooth blow-ups with exceptional divisor given respectively by $V(\texttt{x[38]})$ and $V(\texttt{x[32]})$.

Finally, there is the contractible class

$$\gamma_5: \quad \texttt{x[2] + x[3] + x[7] -2 x[1] = 0}$$

The associated morphism is birational and sends a divisor to a singular point.

Then, we used the command `TabExtremalClasses` to know which contractible classes are extremal and we see that all contractible classes are extremal except two. The non-extremal classes are:

$$\texttt{x[2] - x[3] - x[5] + x[6] = 0}$$
$$\gamma_3: \quad \texttt{x[22] + x[28] - x[32] = 0}$$

Hence, $\mathrm{NE}(X)$ is a cone of dimension 35 with 54 edges. The hyperplane $H_{K_X} \subset \mathcal{N}_1(X)_{\mathbb{Q}}$ of classes which have intersection zero with K_X cuts a facet of $\mathrm{NE}(X)$. Only two edges of $\mathrm{NE}(X)$ do not lie on H_{K_X}: those generated by γ_2 and γ_3.

Observe that γ_2 and γ_3 define smooth blow-ups $X \to X_{\gamma_2} = X_2$ and $X \to X_{\gamma_3} = X_3$. Since γ_2 is extremal and γ_3 is not, X_2 is projective and X_3 is not. Let'compute them using the algorithm `Blowdown`:

```
In[2]:= X2 = Blowdown[X,x[38]];

In[3]:= X3 = Blowdown[X,x[32]];
```

We omit the results of the two commands because the varieties are described by more than 500 primitive relations. Computed X_2 and X_3, we can check that $-K_{X_2}$ and $-K_{X_3}$ are not nef:

```
In[4]:= AnticanonicalNefQ[X2]

Out[4]:= False

In[5]:= AnticanonicalNefQ[X3]

Out[5]:= False
```

7.6 Extremal rays of toric Fano manifolds

In this section we consider the problem of determining the possible number of extremal rays of the Mori cone of a smooth toric Fano variety of dimension $n \leq 4$ and Picard number $\rho \geq 3$.

A smooth complete variety X of dimension n is Fano if its anticanonical bundle $-K_X$ is ample. By definition X is a projective variety.

The Mori cone $\mathrm{NE}(X)$ of a smooth Fano variety X is a closed polyhedral cone of dimension ρ.

If $\mathrm{NE}(X)$ is simplicial, we know that there are exactly ρ extremal rays. However when $\rho \geq 3$ the cone $\mathrm{NE}(X)$ is not always simplicial, and there are no estimations on the number of its extremal rays. We recall that polyhedral cones of dimension 1 or 2 are always simplicial.

Here we restrict our attention to the toric case. There is a finite number of smooth toric Fano varieties for each dimension n. Moreover they are classified up to dimension 4.

In dimension 2 there are five smooth toric Fano varieties: $\mathbb{P}^2$, $\mathbb{P}^1 \times \mathbb{P}^1$ and S_i, the blow-up of $\mathbb{P}^2$ in i points for $i = 1, 2, 3$. They are also called toric Del Pezzo surfaces. In this case the Mori cone is always simplicial except for the variety S_3. Its Mori cone has dimension 4 and 6 extremal rays.

There are 18 smooth toric Fano 3-folds and 13 varieties have $\rho \geq 3$. In this case Batyrev and K. Watanabe & M. Watanabe separately obtain the same classification (see [Ba1], [Ba4] and [WW]).

Smooth toric Fano 4-folds are classified by Batyrev in [Ba4]. In [Sa], Sato describes a toric Fano 4-fold which does not appear in the list given by Batyrev. Both authors use the language of primitive relations to describe them. After these classifications, there are 124 toric Fano 4-folds, among which 114 with $\rho \geq 3$.

We have computed all contractible and extremal classes of every Fano 3-fold and 4-fold using the command `ExtremalClasses` (see Section 4.4). We present here our results.

Let X be a smooth toric Fano variety of dimension at most 4.

After our computations we see that X has no contractible non-extremal classes. Hence, if γ is a contractible class of X and $\varphi_\gamma : X \to X_\gamma$ is the associated contraction, the variety X_γ is projective.

Another remark is that when $\rho = 3$ the Mori cone is always simplicial, hence there are exactly 3 extremal rays.

In dimension 3 there are only two cases with a non-simplicial Mori cone. These ones have maximal Picard number 5 and 7 extremal rays.

Let's consider the case of dimension 4. We sum up our results in the following table. In the table d_2 is the number of toric Fano 4-folds with Picard number ρ, while d_1 is the number of the ones with Picard number ρ whose Mori cone is not simplicial. Moreover $\mathrm{card}_{\mathrm{MAX}}(\mathcal{E})$ is the maximal number of extremal rays among all Fano 4-folds with Picard number ρ.

ρ	d_1/d_2	$\mathrm{card}_{\mathrm{MAX}}(\mathcal{E})$
4	3/47	6
5	9/27	10
6	9/10	20
7	1/1	9
8	1/1	12

Table 7.1: Extremal rays in toric Fano 4-folds with $\rho \geq 4$.

It is interesting to observe that the variety with the maximal number of extremal rays, that is 20, has $\rho = 6$, which is not the maximal Picard number. This one is the Del Pezzo 4-fold (see [VK] and [E1], it is denoted by n. 118 in Table 7.2).

We conclude enumerating all obtained results for the 114 smooth Fano 4-folds with $\rho \geq 3$ in the next table. For each one we list:

1. the name (in most cases we use Batyrev's notation in Section 4 [Ba4]);
2. the Picard number ρ (it is ≥ 3);
3. the cardinality of the set of primitive relations PR;
4. the cardinality of the set of extremal classes $\mathcal{E}$;
5. the CPU-time in seconds (s), it is the CPU-time spent by *Mathematica* to compute the set of extremal classes of the variety.

The symbol (•) denotes the varieties with a non-simplicial Mori cone.

One can observe that the algorithm works quickly. The maximal CPU-time is 0.531 s. It is obtained for the computation of extremal rays of the Del Pezzo toric 4-fold. In dimension 3 the bigger CPU-time spent by *Mathematica* is 0.04 s. It is obtained when `ExtremalClasses` computes the extremal rays of varieties with non-simplicial Mori cone.

Notice that in most cases the CPU-time is equal to 0 s. This means that the command executes some elementary computations which do not involve the use of memory, hence *Mathematica* gives the results easily and quickly.

Variety	ρ	$\mathrm{card}(\mathrm{PR})$	$\mathrm{card}(\mathcal{E})$	CPU-time
D_1	3	3	3	0 s
D_2	3	3	3	0 s
D_3	3	3	3	0 s
D_4	3	3	3	0 s
D_5	3	3	3	0 s
D_6	3	3	3	0 s

Table 7.2(i): Smooth toric Fano 4-folds with $\rho \geq 3$.

Variety	ρ	card(PR)	card($\mathcal{E}$)	CPU-time
D_7	3	3	3	0.011 s
D_8	3	3	3	0 s
D_9	3	3	3	0 s
D_{10}	3	3	3	0 s
D_{11}	3	3	3	0.01 s
D_{12}	3	3	3	0 s
D_{13}	3	3	3	0 s
D_{14}	3	3	3	0 s
D_{15}	3	3	3	0 s
D_{16}	3	3	3	0 s
D_{17}	3	3	3	0 s
D_{18}	3	3	3	0 s
D_{19}	3	3	3	0 s
E_1	3	5	3	0.01 s
E_2	3	5	3	0.01 s
E_3	3	5	3	0.01 s
G_1	3	5	3	0 s
G_2	3	5	3	0 s
G_3	3	5	3	0 s
G_4	3	5	3	0 s
G_5	3	5	3	0.01 s
G_6	3	5	3	0 s
L_1	4	4	4	0 s
L_2	4	4	4	0 s
L_3	4	4	4	0 s
L_4	4	4	4	0 s
L_5	4	4	4	0 s
L_6	4	4	4	0 s
L_7	4	4	4	0 s
L_8	4	4	4	0 s
L_9	4	4	4	0 s
L_{10}	4	4	4	0 s
L_{11}	4	4	4	0 s
L_{12}	4	4	4	0 s
L_{13}	4	4	4	0 s
H_1	4	6	4	0.01 s
H_2	4	6	4	0 s
H_3	4	6	4	0 s
H_4	4	6	4	0.01 s
H_5	4	6	4	0.01 s

Table 7.2(ii): Smooth toric Fano 4-folds with $\rho \geq 3$.

Variety	ρ	$\mathrm{card(PR)}$	$\mathrm{card}(\mathcal{E})$	CPU-time
H_6	4	6	4	0 s
H_7	4	6	4	0.01 s
H_8	4	6	4	0.01 s
H_9	4	6	4	0 s
H_{10}	4	6	4	0 s
I_1	4	6	4	0 s
I_2	4	6	4	0 s
I_3	4	6	4	0.01 s
I_4	4	6	4	0 s
I_5	4	6	4	0 s
I_6	4	6	4	0 s
I_7	4	6	4	0 s
I_8	4	6	4	0 s
I_9	4	6	4	0 s
I_{10}	4	6	4	0.01 s
I_{11}	4	6	4	0 s
I_{12}	4	6	4	0 s
I_{13}	4	6	4	0 s
I_{14}	4	6	4	0 s
I_{15}	4	6	4	0 s
M_1	4	7	4	0 s
M_2	4	7	4	0 s
M_3	4	7	4	0 s
M_4	4	7	4	0 s
M_5	4	7	4	0 s
J_1	4	9	5 (•)	0.03 s
J_2	4	9	6 (•)	0.05 s
Z_1	4	10	4	0 s
Z_2	4	10	5 (•)	0.03 s
Q_1	5	7	5	0.01 s
Q_2	5	7	5	0.01 s
Q_3	5	7	5	0 s
Q_4	5	7	5	0 s
Q_5	5	7	5	0 s
Q_6	5	7	5	0 s
Q_7	5	7	5	0 s
Q_8	5	7	5	0 s
Q_9	5	7	5	0 s
Q_{10}	5	7	5	0 s

Table 7.2(iii): Smooth toric Fano 4-folds with $\rho \geq 3$.

Variety	ρ	$\mathrm{card}(\mathrm{PR})$	$\mathrm{card}(\mathcal{E})$	CPU-time
Q_{11}	5	7	5	0 s
Q_{12}	5	7	5	0 s
Q_{13}	5	7	5	0.01 s
Q_{14}	5	7	5	0 s
Q_{15}	5	7	5	0 s
Q_{16}	5	7	5	0.01 s
Q_{17}	5	7	5	0 s
n. 108	5	10	5	0.01 s
K_1	5	10	7 ($\bullet$)	0.19 s
K_2	5	10	7 ($\bullet$)	0.03 s
K_3	5	10	7 ($\bullet$)	0.04 s
K_4	5	10	7 ($\bullet$)	0.041 s
R_1	5	11 ($\star$)	7 ($\bullet$)	0.05 s
R_2	5	11 ($\star$)	7 ($\bullet$)	0.05 s
R_3	5	11 ($\star$)	7 ($\bullet$)	0.05 s
n. 117	5	14	10 ($\bullet$)	0.14 s
HS ($\diamond$)	5	18	9 ($\bullet$)	0.1 s
$S_2 \times S_2$	6	10	6	0 s
U_1	6	11	8 ($\bullet$)	0.04 s
U_2	6	11	8 ($\bullet$)	0.04 s
U_3	6	11	8 ($\bullet$)	0.04 s
U_4	6	11	8 ($\bullet$)	0.04 s
U_5	6	11	8 ($\bullet$)	0.04 s
U_6	6	11	8 ($\bullet$)	0.041 s
U_7	6	11	8 ($\bullet$)	0.04 s
U_8	6	11	8 ($\bullet$)	0.04 s
n. 118	6	25	20 ($\bullet$)	0.531 s
$S_2 \times S_3$	7	16	9 ($\bullet$)	0.05 s
$S_3 \times S_3$	8	18	12 ($\bullet$)	0.12 s

Table 7.2(iv): Smooth toric Fano 4-folds with $\rho \geq 3$.

($\star$): The varieties R_1, R_2, R_3 have 11 primitive relations but in [Ba4] we find only 10. The missing relation is $v_4 + v_6 = v_0$ (see [Ba4]).

($\diamond$): HS is the variety described by Sato in [Sa].

Appendix A

The package *Toric Varieties*: a user's guide

In this appendix, we present all algorithms of the package *Toric Varieties*.

We refer for the definitions and statements which we have used to construct the algorithms to the previous chapters of this thesis.

A.1 Loading the package

Before starting to work with the package *Toric Varieties* we have to load all commands which we will use.

All programs of package *Toric Varieties* are in the file `toricvar.m`, which is a text file and that you have only to ask *Mathematica* to read it every time that you open a new session.

Mathematica assumes that the files on your computer are organized in a collection of directories, so every time that you use *Mathematica* you have a current working directory and *Mathematica* will load the package only if you specify the directory of the file. If you save the package in the current working directory then the command to use is:

```
<<toricvar.m
```

Instead, if you save the file in another directory you have to specify the full path of file. For example, we use the command:

```
<<"C:\\ToricVarieties\\toricvar.m"
```

In this case, `ToricVarieties` is the name of the directory where we saved the package.

A.2 How to input a variety

Let X be a smooth, complete toric variety of dimension n.

In order to input X in *Mathematica*, you must know either the primitive relations of X or the coordinates of generators with respect to some basis of N together with the maximal cones of the fan.

The easiest way to introduce X is using the command `NewVariety`. We see how it works with an example.

Suppose you want F to be the Hirzebruch surface $\mathbb{F}_1$. You write:

```
In[1]:= F = NewVariety
```

Mathematica opens a window and asks you: `Which is the dimension?`(of the variety). You answer writing `2`, then the *Mathematica* opens another window and asks you: `Which is the number of generators?` (of the fan of variety). You answer `4`.

At this point, you have to decide if you want to describe the variety using the language of primitive relations or the set of generators together with the set of maximal cones. In fact, *Mathematica* asks you: `Do you want to use primitive relations (p) or cones (c) to describe the variety?`

Notice that if t is the number of generators you will have to call them `x[1],...,x[t]`. In the case of $\mathbb{F}_1$ they will be: `x[1], x[2], x[3], x[4]`.

If you answer with `p`, then *Mathematica* opens a window and there you enumerate all primitive relations (as linear polynomials in the generators of the fan):

```
{x[2] + x[3], x[1] + x[4] - x[2]}.
```

Then the algorithm gives its output:

```
Out[1]:= {2, 4, {{{x[2], x[3]}, {{0, 0}}}, {{x[1], x[4]}, {
{{x[2], 1}}}}, Null, Null}
```

Instead, if you answer with `c`, then *Mathematica* asks you the coordinates of generators and the maximal cones. First, you enumerate the coordinates:

```
{x[1] -> {1, 0}, x[2] -> {0, 1}, x[3] -> {0, - 1}, x[4] -> {- 1,
1}}
```

Then, in another window, you enumerate all maximal cones of the fan:

```
{{x[1], x[2]}, {x[1], x[3]}, {x[2], x[4]}, {x[3], x[4]}}
```

In this case the output of the algorithm is:

```
Out[1]:= {2, 4, Null, {x[1] -> {1, 0}, x[2] -> {0, 1}, x[3] -> {0,
- 1}, x[4] -> {- 1, 1}}, {{x[1], x[2]}, {x[1], x[3]}, {x[2],
x[4]}, {x[3], x[4]}}}
```

Remark A.2.1. In some case the command `NewVariety` could answer with the following error message:

```
   Syntax:: sntxi:
Incomplete expression; more input is needed. More...
```

This means that the data which you want to enumerate is given by a list of symbols with a number k of characters and $k > 252$. You can resolve the problem defining a list before using the command `NewVariety`.

Suppose that you want to describe a variety X with the language of primitive relations. Then you can define your variety proceeding in the following way:

1. you define a list L where you write all primitive relations as polynomials in the generators;
2. you use the command `NewVariety` to define X and when the algorithm asks you the primitive relations you write in the opened window the name of the list L.

At the end you obtain the list which describes the variety.

Notice that if you want to describe X with the data of maximal cones and generators you have to define two lists: the list of generators and the list of maximal cones. Then you have to write their names in the windows that the algorithm opens.

Besides `NewVariety`, there are three commands to introduce specific varieties:

- `ProjSpace[n]`, it gives the `n`-dimensional projective space;
- `DelPezzo[n]`, it gives the `n`-dimensional Del Pezzo variety;
- `PseudoDelPezzo[n]`, it gives the `n`-dimensional pseudo Del Pezzo variety.

Del Pezzo and pseudo Del Pezzo varieties are Fano varieties introduced by V. E. Voskresenskiĭ and Alexander Klyachko in [VK] and by Ewald in [E1] to prove some statements on Fano toric varieties. Their dimension is even so `n` must be even.

Let's consider in more detail the way of representing a toric variety X in *Mathematica*. In any case, the output of `Newvariety` is a list of five elements:

$$\{n, t, \mathrm{PR}, \mathrm{G}, \mathrm{MC}\} \tag{A.1}$$

where:

- n is the dimension of variety;
- t is the cardinality of the set of generators of fan associated to X;

- PR is the set of primitive relations;
- G is the set of the coordinates of the generators with respect to some basis of N;
- MC is the set of maximal cones.

A primitive relation of the form:

$$\mathtt{x[i_1]+\cdots+x[i_k]-(a_1x[j_1]+\cdots+a_hx[i_h])=0}$$

is represented in PR as the list:

$$\mathtt{\{\{x[i_1],\ldots,x[i_k]\},\ \{\{x[j_1],a_1\},\ldots,\{x[i_h],a_h\}\}} \tag{A.2}$$

A toric variety X is completely determined by the data $n, t, \mathrm{G}, \mathrm{MC}$. In the cases of a projective variety and of a variety which has a projective blow-up, you can use also the data n, t, PR to determine X completely (see Chapter 3). Hence, the list (A.1) must contain at least n, t, PR or $n, t, \mathrm{G}, \mathrm{MC}$. *Mathematica* will replace the missing entries with `Null`.

If you have a list with a `Null` entry and you want to know all elements of the list $\{n, t, \mathrm{PR}, \mathrm{G}, \mathrm{MC}\}$, you can use the command `AddInput[X]`.

Let F be the Hirzebruch surface $\mathbb{F}_1$:

```
In[1]:= F = {2, 4, Null, {x[1] -> {1, 0}, x[2] -> {0, 1}, x[3] ->
{0, - 1}, x[4] -> {- 1, 1}}, {{x[1], x[2]}, {x[1], x[3]}, {x[2],
x[4]}, {x[3], x[4]}}};
```

Here you know the coordinates of the generators and the cones of the fan.

Suppose that you want to know which primitive relations characterize the variety, then you write:

```
In[2]:= AddInput[F]

Out[2]:= {2, 4, {{{x[2], x[3]}, {{0, 0}}}, {{x[1], x[4]}, {
{{x[2], 1}}}}, {x[1] -> {1, 0}, x[2] -> {0, 1}, x[3] -> {0, - 1},
x[4] -> {- 1, 1}}, {{x[1], x[2]}, {x[1], x[3]}, {x[2], x[4]},
{x[3], x[4]}}}
```

Instead, if the Hirzebruch surface $\mathbb{F}_1$ is described by the language of primitive relations:

```
In[1]:= {2, 4, {{{x[2], x[3]}, {{0, 0}}}, {{x[1], x[4]}, {{x[2],
1}}}}, Null, Null}
```

Then you can know the list of the coordinates of generators and the set of maximal cones writing:

```
In[2]:= AddInput[F]

Out[2]:= {2, 4, {{{x[2], x[3]}, {{0, 0}}}, {{x[1], x[4]}, {
{{x[2], 1}}}}}, {x[1] -> {1, 0}, x[2] -> {0, 1}, x[3] -> {0, - 1},
x[4] -> {- 1, 1}}, {{x[1], x[2]}, {x[1], x[3]}, {x[2], x[4]},
{x[3], x[4]}}}
```

We specify that you can also describe the surface with the list $\{n, t, \mathrm{PR}, \mathrm{G}, \mathrm{Null}\}$. In this case, the algorithm `AddInput` adds only the list of maximal cones.

A.3 Describing cones in the fan

Cone Test

`ConeQ[X, set]` allows to know if a subset of generators of the fan of X spans a cone in the fan.

The input of the algorithm are:

1. the variety X;
2. a subset of the generators of the fan.

The output is `True` if the subset generates a cone in the fan, `False` otherwise.

Example A.3.1. Let F be the Hirzebruch surface $\mathbb{F}_1$:

```
In[1]:= F = {2, 4, {{{x[2], x[3]}, {{0, 0}}}, {{x[1], x[4]}, {
{{x[2],1}}}}}, Null, Null};

In[2]:= ConeQ[F, {x[1], x[2]}]

Out[2]:= True

In[3]:= ConeQ[F, {x[1], x[4]}]

Out[3]:= False
```

Cones of dimension k

The algorithm `Cones[X, k]` gives the list of all `k`-dimensional cones in the fan of X.

The input are:

1. the variety X;
2. the dimension k.

The output is the list of the sets of generators of k-dimensional cones.

Example A.3.2. Let F be the Hirzebruch surface $\mathbb{F}_1$ as in Example A.3.1:

```
In[1]:=  Cones[F, 1]

Out[1]:= {{x[1]}, {x[2]}, {x[3]}, {x[4]}}

In[2]:=  Cones[F, 2]

Out[2]:= {{x[1], x[2]}, {x[1], x[3]}, {x[2], x[4]}, {x[3], x[4]}}
```

A.4 Primitive relations and contractible classes

The table of primitive relations

The algorithm TabVar[X] gives a table in which we enumerate all primitive relations.

The input is the variety X given by the list of primitive relations.

The output of the program is a table of all primitive relations of the variety, given as linear polynomial in the generators. If a primitive relation is contractible the program marks the fact putting the letter C after the primitive relation.

Example A.4.1. Let T be the Oda 3-fold. T is a smooth complete toric 3-fold with Picard number 4. It is not projective. A detailed description of T can be found on page 84 in [O2].

```
In[1]:= T = {3, 7, {{{x[1], x[4]}, {{x[7], 1}}}, {{x[1], x[5]},
{{x[2], 1}, {x[7], 1}}}, {{x[2], x[4]}, {{x[5], 1}}}, {{x[2],
x[6]}, {{x[3], 1}, {x[5], 1}}}, {{x[3], x[4]}, {{x[6], 1}}},
{{x[3], x[7]}, {{x[1], 1}, {x[6], 1}}}, {{x[5], x[6], x[7]},
{{x[4], 2}}}}, Null, Null};

In[2]:= TabVar[T]

Out[2]:=

 x[1] + x[4]          - x[7]          =  0
 x[1] + x[5]          - x[2] - x[7]   =  0  C
 x[2] + x[4]          - x[5]          =  0
 x[2] + x[6]          - x[3] - x[5]   =  0  C
 x[3] + x[4]          - x[6]          =  0
 x[3] + x[7]          - x[1] - x[6]   =  0  C
 x[5] + x[6] + x[7]   -2 x[4]         =  0  C
```

The table of contractible primitive relations

The algorithm `TabContrRel[X]` is similar, but it lists only the contractible primitive relations.

The input is the variety described by primitive relations.

The output is the table of the contractible primitive relations of the variety.

Example A.4.2. Let T be the Oda 3-fold as in Example A.4.1:

```
In[1]:= TabContrRel[T]

Out[1]:=

 x[1] + x[5]          - x[2] - x[7]  =  0
 x[2] + x[6]          - x[3] - x[5]  =  0
 x[3] + x[7]          - x[1] - x[6]  =  0
 x[5] + x[6] + x[7]  -2 x[4]         =  0
```

A.5 Invariant curves

Relation of an invariant curve

The algorithm `Curve[X, cone]` gives the relation associated to the numerical class of an invariant curve C in X.

The input are:

1. the variety X;
2. the $(n-1)$-dimensional cone corresponding to C given by the list of its generators.

The output of the algorithm is a linear polynomial whose variables are the generators of the fan of X.

Example A.5.1. Let T be the Oda 3-fold as in Example A.4.1. The 2-dimensional cones of T are:

```
In[1]:= Cones[T, 2]

Out[1]:= {{x[1], x[2]}, {x[1], x[3]}, {x[1], x[6]}, {x[1], x[7]},
{x[2], x[3]}, {x[2], x[5]}, {x[2], x[7]}, {x[3], x[5]}, {x[3],
x[6]}, {x[4], x[5]}, {x[4], x[6]}, {x[4], x[7]}, {x[5], x[6]},
{x[5], x[7]}, {x[6], x[7]}}
```

Then, for every 2-cone we can determine the relation associated to the class (modulo numerical equivalence) of the corresponding invariant curve. For example, you can compute the relation associated to the numerical class of the invariant curve $C = V(\langle \texttt{x[1],x[6]} \rangle)$, writing:

```
In[2]:= Curve[T, {x[1],x[6]}]

Out[2]:= -x[1] + x[3] - x[6] + x[7]
```

We observe that if the variety X is described by the list of generators and maximal cones then the algorithm gives its answer more quickly.

The table of all invariant curves

The algorithm `TabInvCurves[X]` enumerates all invariant curves of the variety X in a table.

The input is the variety X.

The output is a table of two columns:

1. in the first there is the cone which represents the invariant curve, it is given by the list of its generators;
2. in the second there is the linear polynomial in the generators of the fan corresponding to the numerical class associated to the invariant curve.

Example A.5.2. Let T be the Oda 3-fold represented by the list:

```
In[1]:= T = {3, 7, Null, {x[1] -> {1, 0, 0}, x[2] -> {0, 1, 0},
x[3] -> {0, 0, 1}, x[4] -> {-1, -1, -1}, x[5] -> {-1, 0, -1}, x[6]
-> {-1, -1, 0}, x[7] -> {0, -1, -1}}, {{x[1], x[2], x[3]}, {x[1],
x[2], x[7]}, {x[1], x[3], x[6]}, {x[1], x[6], x[7]}, {x[2], x[3],
x[5]}, {x[2], x[5], x[7]}, {x[3], x[5], x[6]}, {x[4], x[5], x[6]},
{x[4], x[5], x[7]}, {x[4], x[6], x[7]}}};

In[2]:= TabInvCurves[T]

Out[2]:=
```

x[1]x[2]	x[2] + x[3] + x[7]	=	0
x[1]x[3]	x[1] + x[2] + x[6]	=	0
x[1]x[6]	-x[1] + x[3] - x[6] + x[7]	=	0
x[1]x[7]	x[1] + x[2] + x[6]	=	0
x[2]x[3]	x[1] + x[3] + x[5]	=	0
x[2]x[5]	x[2] + x[3] + x[7]	=	0
x[2]x[7]	x[1] - x[2] + x[5] - x[7]	=	0
x[3]x[5]	x[2] - x[3] - x[5] + x[6]	=	0
x[3]x[6]	x[1] - x[3] + x[5]	=	0
x[4]x[5]	-2x[4] + x[5] + x[6] + x[7]	=	0
x[4]x[6]	-2x[4] + x[5] + x[6] + x[7]	=	0
x[4]x[7]	-2x[4] + x[5] + x[6] + x[7]	=	0
x[5]x[6]	x[3] + x[4] - x[6]	=	0
x[5]x[7]	x[2] + x[4] - x[5]	=	0
x[6]x[7]	x[1] + x[4] - x[7]	=	0

The representation of a numerical class of a curve in $\mathcal{N}_1(X)_{\mathbb{Q}}$

Given X, you can consider the set of all relations associated to all different numerical classes of invariant curves. Every relation is an element of the vector space $\mathcal{N}_1(X)_{\mathbb{Q}}$. Moreover, the set contains a basis B of $\mathcal{N}_1(X)_{\mathbb{Q}}$.

The algorithm determines a basis B of $\mathcal{N}_1(X)_{\mathbb{Q}}$ and computes the coordinates of every element of the set with respect to B.

The input is the variety X.

The output is a table. If `a[i1]x[i1] +···+ a[ik]x[ik]` is the relation associated to a numerical class of an invariant curve C and `{b1, ..., bρ}` are its coordinates, then, in the table, you find the element:

$$\texttt{a[}i_1\texttt{]x[}i_1\texttt{]} + \cdots + \texttt{a[}i_k\texttt{]x[}i_k\texttt{]} \texttt{ -> } b_1 \ldots b_\rho$$

Example A.5.3. Let T be the Oda 3-fold as in the Example A.4.1:

```
In[1]:=  CoordCurvesClasses[T]

Out[1]:=

 x[2] + x[4] - x[5]                -> 1 0 0 0
 x[1] + x[3] + x[5]                -> 2 2 1 1
 x[3] + x[4] - x[6]                -> 1 1 1 0
 x[1] + x[2] + x[6]                -> 2 1 0 1
 x[2] - x[3] - x[5] + x[6]         -> 0 -1 -1 0
 x[1] + x[4] - x[7]                -> 1 1 0 0
 x[1] - x[2] + x[5] - x[7]         -> 0 1 0 0
 x[2] + x[3] + x[7]                -> 2 1 1 1
 -x[1] + x[3] - x[6] + x[7]        -> 0 0 1 0
 -2 x[4] + x[5] + x[6] + x[7]      -> 0 0 0 1
```

A.6 Blowing up

The algorithm `Blowup[X, cone]` computes the blow-up Y of the variety X along an invariant subvariety V. The subvariety V is given by the corresponding cone in Σ_X.

The input are:

1. the variety X;
2. the list of generators of the cone associated to V.

The output is a list of type (A.1), which represents Y.

The next example clarifies how the algorithm works.

Example A.6.1. In this example we construct a projective variety W whose blow-down is the Oda 3-fold. W is obtained blowing up a $\mathbb{P}^3$ along four curves:

```
In[1]:= Z = ProjSpace[3];

Out[1]:= TabVar[Z]

 x[1] + x[2] + x[3] + x[4]  =  0  C

In[2]:= Z1 = Blowup[Z,{x[2], x[4]}];

Out[2]:= TabVar[Z1]

 x[1] + x[3] + x[5]          =  0  C
 x[2] + x[4]         - x[5]  =  0  C

In[3]:= Z2 = Blowup[Z1,{x[3], x[4]}];

Out[3]:= TabVar[Z2]

 x[2] + x[4]         - x[5]          =  0
 x[2] + x[6]         - x[3] - x[5]  =  0  C
 x[3] + x[4]         - x[6]          =  0  C
 x[1] + x[3] + x[5]                  =  0
 x[1] + x[5] + x[6]  - x[4]          =  0  C

In[4]:= Z3 = Blowup[Z2,{x[1], x[4]}];

Out[4]:= TabVar[Z3]

 x[1] + x[4]         - x[7]          =  0  C
 x[2] + x[4]         - x[5]          =  0
 x[2] + x[6]         - x[3] - x[5]  =  0  C
 x[2] + x[7]         - x[1] - x[5]  =  0  C
 x[3] + x[4]         - x[6]          =  0
 x[3] + x[7]         - x[1] - x[6]  =  0  C
 x[1] + x[3] + x[5]                  =  0
 x[1] + x[5] + x[6]  - x[4]          =  0
 x[5] + x[6] + x[7]  -2 x[4]         =  0  C

In[5]:= W = Blowup[Z3,{x[1], x[5]}];

Out[5]:= TabVar[W]
```

```
x[1] + x[4]         - x[7]          =  0
x[1] + x[5]         - x[8]          =  0  C
x[2] + x[4]         - x[5]          =  0
x[2] + x[6]         - x[3] - x[5]   =  0  C
x[2] + x[7]         - x[8]          =  0  C
x[3] + x[4]         - x[6]          =  0
x[3] + x[7]         - x[1] - x[6]   =  0  C
x[3] + x[8]                         =  0
x[4] + x[8]         - x[5] - x[7]   =  0  C
x[6] + x[8]         - x[4]          =  0
x[5] + x[6] + x[7]  -2 x[4]         =  0  C
```

Notice that the output of algorithm can have four forms:

1. if X is described by the list: $X = \{n, t, \mathrm{PR}^X, \mathrm{Null}, \mathrm{Null}\}$, then the blowup is given by the list $Y = \{n, t+1, \mathrm{PR}^Y, \mathrm{Null}, \mathrm{Null}\}$;
2. if X is described by the list: $X = \{n, t, \mathrm{PR}^X, \mathrm{G}^X, \mathrm{Null}\}$, then the blowup is given by the list $Y = \{n, t+1, \mathrm{PR}^Y, \mathrm{G}^Y, \mathrm{Null}\}$;
3. if X is described by the list: $X = \{n, t, \mathrm{PR}^X, \mathrm{G}^X, \mathrm{MC}^X\}$, then the blowup is given by the list $Y = \{n, t+1, \mathrm{PR}^Y, \mathrm{G}^Y, \mathrm{MC}^Y\}$;
4. if X is described by the list of maximal cones, then $Y = \{n, t+1, \mathrm{Null}, \mathrm{G}^Y, \mathrm{MC}^Y\}$.

A.7 Blowing down

Suppose that the fan of Y has a contractible primitive relation of type:

$$\texttt{x[1]} + \cdots + \texttt{x[k]} - \texttt{x[h]} = 0.$$

Then, the contraction of this class is a blow-down $\psi : Y \to X$ whose exceptional divisor is $V($`x[h]`$)$. The variety X can be computed from Y with the command `Blowdown[Y, x[h]]`.

Observe that we are only specifying the exceptional divisor of the blowdown. If there is more than one contractible primitive relation of type `x[1]` $+\cdots+$ `x[k] - x[h] = 0`, then *Mathematica* shows the possible choices and asks you to choose which relation you want to contract.

The input are:

1. the variety Y;
2. the generator `x[h]`.

The output is a list of type (A.1), which represents Y.

We show how the algorithm works in the next examples.

Example A.7.1. Let W be the variety which we have constructed in Example A.6.1. It determines the Oda 3-fold when you consider its blow-down with respect to the primitive relation `x[2]+x[7]-x[8]=0`.

If you try to contract the generator `x[8]` and you have to write:

```
In[1]:= T = Blowdown[W, x[8]]
```

Since there are two possible contractible primitive relations, *Mathematica* asks you which primitive relation it has to contract. *Mathematica* opens a window and shows the following message:

```
    {Which is the primitive relation to contract, (give the  number)?,
    {x[1] + x[5] - x[8], x[2] + x[7] - x[8]}}
```

If you answer to the question with the number `2` then the algorithm computes the blow-down:

```
Out[1]:= {3, 7, {{{x[1], x[4]}, {{x[7], 1}}}, {{x[1], x[5]},
{{x[2], 1}, {x[7], 1}}}, {{x[2], x[4]}, {{x[5], 1}}}, {{x[2],
x[6]}, {{x[3], 1}, {x[5], 1}}}, {{x[3], x[4]}, {{x[6], 1}}},
{{x[3], x[7]}, {{x[1], 1}, {x[6], 1}}}, {{x[5], x[6], x[7]},
{{x[4], 2}}}}, {x[1] -> {1, 0, 0}, x[2] -> {0, 1, 0}, x[3] -> {0,
0, 1}, x[4] -> {-1, -1, -1}, x[5] -> {-1, 0, -1}, x[6] -> {-1, -1,
0}, x[7] -> {0, -1, -1}}, {{x[1], x[2], x[3]}, {x[1], x[2], x[7]},
{x[1], x[3], x[6]}, {x[1], x[6], x[7]}, {x[2], x[3], x[5]}, {x[2],
x[5], x[7]}, {x[3], x[5], x[6]}, {x[4], x[5], x[6]}, {x[4], x[5],
x[7]}, {x[4], x[6], x[7]}}}
```

```
In[2]:= TabVar[T]
```

```
Out[2]:=

 x[1] + x[4]          - x[7]          =  0
 x[1] + x[5]          - x[2] - x[7]   =  0  C
 x[2] + x[4]          - x[5]          =  0
 x[2] + x[6]          - x[3] - x[5]   =  0  C
 x[3] + x[4]          - x[6]          =  0
 x[3] + x[7]          - x[1] - x[6]   =  0  C
 x[5] + x[6] + x[7]   -2 x[4]         =  0  C
```

This is the Oda 3-fold.

The algorithm is able to compute the blow-down also when the variety is described by the list $\{n, t, \text{Null}, \text{G}^Y, \text{MC}^Y\}$.

Example A.7.2. Let W be the variety constructed in Example A.6.1 and suppose that it is described only by the maximal cones and generators:

```
In[1]:= W = {3, 8, Null, {x[1] -> {1, 0, 0}, x[2] -> {0, 1, 0},
```

```
x[3] -> {0, 0, 1}, x[4] -> {-1, -1, -1}, x[5] -> {-1, 0, -1}, x[6]
-> {-1, -1, 0}, x[7] -> {0, -1, -1}, x[8] -> {0, 0, -1}}, {{x[1],
x[2], x[3]}, {x[1], x[2], x[8]}, {x[1], x[3], x[6]}, {x[1], x[6],
x[7]}, {x[1], x[7], x[8]}, {x[2], x[3], x[5]}, {x[2], x[5], x[8]},
{x[3], x[5], x[6]}, {x[4], x[5], x[6]}, {x[4], x[5], x[7]}, {x[4],
x[6], x[7]}, {x[5], x[7], x[8]}}};
```

Suppose that you want to construct its blow-down with respect to `x[8]`:

```
In[2]:= Blowdown[W, x[8]]
```

Then *Mathematica* opens a window and asks you:

```
Which is the contractible primitive relation associated to
the blow-down?
```

You answer writing the polynomial: `x[2] + x[7] - x[8]` and then *Mathematica* computes the blow-down:

```
Out[2]:= {3, 7, Null, {x[1] -> {1, 0, 0}, x[2] -> {0, 1, 0}, x[3]
-> {0, 0, 1}, x[4] -> {-1, -1, -1}, x[5] -> {-1, 0, -1}, x[6] ->
{-1, -1, 0}, x[7] -> {0, -1, -1}}, {{x[1], x[2], x[3]}, {x[1],
x[2], x[7]}, {x[1], x[3], x[6]}, {x[1], x[6], x[7]}, {x[2], x[3],
x[5]}, {x[2], x[5], x[7]}, {x[3], x[5], x[6]}, {x[4], x[5], x[6]},
{x[4], x[5], x[7]}, {x[4], x[6], x[7]}}}
```

Notice that in this case the algorithm is not able to test if the relation is contractible so it remembers you asking for a contractible primitive relation.

Observe that the output of algorithm can have four forms:

1. if Y is described by the list: $Y = \{n, t, \mathrm{PR}^Y, \mathrm{Null}, \mathrm{Null}\}$, then the blow-down is given by the list $X = \{n, t-1, \mathrm{PR}^X, \mathrm{Null}, \mathrm{Null}\}$;
2. if Y is described by the list: $Y = \{n, t, \mathrm{PR}^Y, \mathrm{G}^Y, \mathrm{Null}\}$, then the blow-down is given by the list $X = \{n, t-1, \mathrm{PR}^X, \mathrm{G}^X, \mathrm{Null}\}$;
3. if Y is described by the list: $Y = \{n, t, \mathrm{PR}^Y, \mathrm{G}^Y, \mathrm{MC}^Y\}$, then the blow-down is given by the list $X = \{n, t-1, \mathrm{PR}^X, \mathrm{G}^X, \mathrm{MC}^X\}$;
4. if $Y = \{n, t, \mathrm{Null}, \mathrm{G}^Y, \mathrm{MC}^Y\}$ is described by the list of maximal cones and generators, then the blow-down is $X = \{n, t-1, \mathrm{Null}, \mathrm{G}^X, \mathrm{MC}^X\}$.

A.8 Projectivity test

The algorithm `ProjQ[X]` says if the variety X is projective or not.

The input is the variety X.

The output is a string: `True` if the variety is projective, `False` otherwise.

Example A.8.1. Let T be the Oda 3-fold as in the Example A.5.2:

```
In[1]:=  ProjQ[T]

Out[1]:= False

In[2]:= W = Blowup[T, {x[2],x[7]}];
```

W is the variety described in Example A.6.1. It is projective, in fact:

```
In[3]:= ProjQ[W]

Out[3]:= True
```

A.9 Extremal classes

The table of extremal and non-extremal classes

Let X be a projective variety. Suppose that you want to describe the Mori cone $\mathrm{NE}(X) \subset \mathcal{N}_1(X)_{\mathbb{Q}}$. The algorithm `TabExtremalClasses[X]` determines the extremal and non-extremal classes in the set of contractible primitive relations.

The input of the algorithm is a projective variety X.

The output is the table of all contractible classes, together with their coordinates with respect to a basis of $\mathcal{N}_1(X)_{\mathbb{Q}}$ (the basis is chosen among the contractible classes).

The classes marked with the letter `E` are the extremal ones.

The following example shows how the algorithm works.

Example A.9.1. Let W be the variety described in Example A.6.1:

```
In[1]:=  TabExtremalClasses[W]

Out[1]:=

 x[2] - x[3] - x[5] + x[6]      ->  -1 1 -1 0 0  E
 x[1] + x[5] - x[8]             ->  1 0 0 0 0    E
 -x[1] + x[3] - x[6] + x[7]     ->  0 0 1 0 0    E
 x[4] - x[5] - x[7] + x[8]      ->  0 0 0 1 0    E
 -2 x[4] + x[5] + x[6] + x[7]   ->  0 0 0 0 1    E
 x[2] + x[7] - x[8]             ->  0 1 0 0 0
```

In this case there is only one non-extremal class, whose blow-down gives the Oda 3-fold.

Example A.9.2. Let S be the following variety:

```
In[2]:= S = {{7, 12, {{{x[3], x[12]}, {{x[4], 1}, {x[5], 2},
{x[9], 1}}}, {{x[1], x[2], x[3]}, {{x[9], 1}}}, {{x[1], x[3],
x[11]}, {{x[4], 1}, {x[6], 1}, {x[9], 1}}}, {{x[1], x[4], x[5]},
{{x[10], 1}}}, {{x[1], x[4], x[12]}, {{x[2], 1}, {x[10], 2}}},
{{x[1], x[5], x[11]}, {{x[2], 1}, {x[6], 1}, {x[10], 1}}}, {{x[1],
x[11], x[12]}, {{x[2], 2}, {x[6], 1}, {x[10], 2}}}, {{x[2], x[3],
x[10]}, {{x[4], 1}, {x[5], 1}, {x[9], 1}}}, {{x[2], x[4], x[6]},
{{x[11], 1}}}, {{x[2], x[5], x[10]}, {{x[12], 1}}}, {{x[3], x[10],
x[11]}, {{x[4], 2}, {x[5], 1}, {x[6], 1}, {x[9], 1}}}, {{x[4],
x[6], x[12]}, {{x[5], 1}, {x[10], 1}, {x[11], 1}}}, {{x[5], x[7],
x[8], x[9], x[11]}, {{x[2], 1}}}, {{x[6], x[7], x[8], x[9],
x[10]}, {{x[1], 1}}}, {{x[6], x[7], x[8], x[9], x[12]}, {{x[1],
1}, {x[2], 1}, {x[5], 1}}}, {{x[7], x[8], x[9], x[10], x[11]},
{{x[1], 1}, {x[2], 1}, {x[4], 1}}}, {{x[7], x[8], x[9], x[11],
x[12]}, {{x[2], 2}, {x[10], 1}}}, {{x[4], x[5], x[6], x[7], x[8],
x[9]}, {{0, 0}}}}}, Null, Null};
```

S is a projective variety, has dimension 7 and Picard number 5.

```
In[3]:= TabExtremalClasses[S]

Out[3]:=

 -x[1] -x[2] -x[4] +x[7] +x[8] +x[9] +x[10] +x[11]  ->  0 0 0 -1 1  E
 x[1] -2x[2] -x[6] -2x[10] +x[11] +x[12]            ->  1 0 0 0 0   E
 x[2] + x[5] +x[10] -x[12]                          ->  0 1 0 0 0   E
 x[3] -2x[4] -x[5] -x[6] -x[9] +x[10] +x[11]        ->  0 0 1 0 0   E
 x[4] -x[5] +x[6] -x[10] -x[11] +x[12]              ->  0 0 0 1 0   E
 -x[1] -x[2] -x[5] +x[6] +x[7] +x[8] +x[9] +x[12]   ->  0 0 0 0 1
```

In this case there are five extremal classes and one non-extremal contractible class.

The coordinates of contractible primitive relations in $\mathcal{N}_1(X)_{\mathbb{Q}}$

The algorithm `CoordContrClasses[X]` is similar to the algorithm `CoordCurvesClasses[X]`, but it works only with the contractible classes of curves.

Let X be a projective variety. Consider the set $\mathcal{C}$ of all contractible primitive relations. This set contains a basis B of $\mathcal{N}_1(X)_{\mathbb{Q}}$. The algorithm determines the basis B and computes the coordinates of every element with respect to B.

The input is the variety X.

The output is the table of all contractible classes, together with their coordinates with respect to a basis of $\mathcal{N}_1(X)_{\mathbb{Q}}$ (the basis is chosen among the contractible classes).

The following example shows how the algorithm works.

Example A.9.3. Let W the variety of Example A.6.1:

```
In[1]:=  CoordContrClasses[W]
```

```
Out[1]:=
```

```
 x[1] + x[5] - x[8]              ->  1 0 0 0 0
 x[2] - x[3] - x[5] + x[6]       ->  -1 1 -1 0 0
 x[2] + x[7] - x[8]              ->  0 1 0 0 0
 - x[1] + x[3] - x[6] + x[7]     ->  0 0 1 0 0
 x[4] - x[5] - x[7] + x[8]       ->  0 0 0 1 0
 -2 x[4] + x[5] + x[6] + x[7]    ->  0 0 0 0 1
```

A.10 Isomorphism test

The algorithm `IsomQ[X, Y]` says if the two varieties X and Y are isomorphic.

The input are the varieties X and Y.

The output is the string `True` when X and Y are isomorphic, `False` otherwise.

The next example explains how the algorithm works.

Example A.10.1. Let F be the Hirzebruch surface $\mathbb{F}_1$:

```
In[1]:= F = {2, 4, {{{x[3], x[4]}, {{0, 0}}}, {{x[1], x[2]},
{{x[4], 3}}}}, Null, Null};
```

Let H be again the Hirzebruch surface $\mathbb{F}_1$, given by a different but isomorphic fan:

```
In[2]:= H = {2, 4, {{{x[5], x[2]}, {{0, 0}}}, {{x[1], x[4]},
{{x[2], 3}}}}, Null, Null};
```

They are isomorphic and the answer of algorithm confirms our assertion:

```
In[3]:= IsomQ[F, H]
```

```
Out[3]:= True
```

The algorithm tests if two varieties are isomorphic, it works applying Proposition 3.1.1 in Chapter 3, so the answer can be wrong in the case of a smooth, complete, non-projective toric variety.

A.11 Products

Given two smooth, complete, toric varieties X and Y, their product is again a smooth, complete, toric variety Z. The algorithm `Prod[X,Y]` gives a representation of the variety Z obtained as the product of X and Y.

The input are the two varieties X and Y.

The output of the algorithm is the variety Z and the description of Z depends on the description of X and Y. If X and Y have the same representation, that is, if both the two varieties are described by primitive relations (respectively by maximal cones), then Z will be described by primitive relations (respectively by maximal cones).

Instead, if X is given by the data of maximal cones and Y by primitive relations, then you can decide which representation you want for Z. *Mathematica* asks you: `Do you want to use either primitive relations (p) or cones (c) to describe the variety?`.

If you answer `c`, then the solution Z will be described by using maximal cones, if you answer `p` then Z will be described by primitive relations.

We give an example:

Example A.11.1. In order to input $Z = \mathbb{P}^1 \times \mathbb{P}^2$, we write:

```
In[1]:= Z = Prod[ProjSpace[1],ProjSpace[2]]

Out[1]:= {3, 5, {{{x[1], x[2]}, {{0, 0}}}, {{x[3], x[4], x[5]},
{{0, 0}}}}, {x[1] -> {1, 0, 0}, x[2] -> {-1, 0, 0}, x[3] -> {0, 1,
0}, x[4] -> {0, 0, 1}, x[5] -> {0, -1, -1}}, {{x[1], x[3], x[4]},
{x[1], x[3], x[5]}, {x[1], x[4], x[5]}, {x[2], x[3], x[4]}, {x[2],
x[3], x[5]}, {x[2], x[4], x[5]}}}
```

A.12 Fano and anticanonical nef test

The algorithm `FanoQ[X]` says if a variety X is Fano or not. The algorithm `AnticanonicalNefQ[X]` tests if the anticanonical bundle is nef.

The input is the variety X.

The output is the string `True` when X is Fano (when the anticanonical bundle is nef), `False` otherwise.

Example A.12.1. Let F_2 be the Hirzebruch surface $\mathbb{F}_2$:

```
In[1]:= F2 = {2, 4, {{{x[1], x[2]}, {{0, 0}}}, {{x[3], x[4]},
{{x[1], 2}}}}, Null, Null};
```

F_2 is not Fano toric variety, in fact the algorithm `FanoQ` answers negatively:

```
In[2]:= FanoQ[F2]
```

```
Out[2]:= False
```

Its anticanonical bundle is nef, in fact the algorithm `AnticanonicalNefQ` answers:

```
In[3]:= AnticanonicalNefQ[F2]
```

```
Out[3]:= True
```

A.13 Technical commands

A.13.1 Algorithms studying properties of the fan and the lattice N

Generators

The algorithm `Gen[X]` gives the set of generators of fan of variety X.

The input is the variety X.

The output is the list of generators of the fan of the variety X.

Example A.13.1. Let $Z = \mathbb{P}^2 \times \mathbb{P}^3$ be:

```
In[1]:= Z = {5, 7, {{{x[1], x[2], x[3]}, {{0, 0}}}, {{x[4], x[5],
x[6], x[7]}, {{0, 0}}}}, Null, Null}
```

```
In[2]:= Gen[Z]
```

```
Out[2]:= {x[1], x[2], x[3], x[4], x[5], x[6], x[7]}
```

Computing a basis of N

The algorithm `Basis[X]` computes a basis of the lattice N, contained in the set of generators of the fan of X.

The input is the variety X.

The output is a list of elements of type `x[i]`.

Example A.13.2. Let $Z = \mathbb{P}^2 \times \mathbb{P}^3$ be:

```
In[1]:= Z = {5, 7, {{{x[1], x[2], x[3]}, {{0, 0}}}, {{x[4], x[5],
x[6], x[7]}, {{0, 0}}}}, Null, Null}
```

```
In[2]:= Basis[Z]
```

```
Out[2]:= {x[1], x[2], x[4], x[5], x[6]}
```

In general, this algorithm is used when X is described by primitive relations, because when a variety is described by maximal cones, it is known that the set of generators of every maximal cone is a basis for the lattice N.

Basis test

The algorithm `BasisQ[X, set]` says if a subset of generators of the fan of X is a basis for N.

The input are:

1. the variety X;
2. a subset of generators of fan of X.

The output is `True` if the set is a basis, `False` otherwise.

Example A.13.3. Let F_2 be the Hirzebruch surface $\mathbb{F}_2$ as in Example A.12.1:

```
In[1]:= BasisQ[F2, {x[1], x[4]}]

Out[1]:= True

In[2]:= BasisQ[F2, {x[3], x[4]}]

Out[2]:= False
```

Computing a basis of N and the coordinates of generators of the fan of X

`BasisCoord[X]` computes a basis of the lattice N, contained in the set of generators of the fan of X and the coordinates of all generators of Σ_X with respect to this basis.

The input is the variety X.

The output is a list of two sets:

1. the first set is a basis of the lattice N;
2. the second set is the list of all coordinates of every generators, every element has the form:

$$\texttt{x[i]} \ \texttt{->} \ \{\texttt{x[i]}_1,\ldots,\texttt{x[i]}_n\}.$$

Example A.13.4. Let F_2 be the Hirzebruch surface $\mathbb{F}_2$ as in Example A.12.1:

```
In[1]:= BasisCoord[F2]

Out[1]:= {{x[1], x[3]}, {x[1] -> {1, 0}, x[2] -> {-1, 0}, x[3] ->
{0, 1}, x[4] -> {2, -1}}}
```

In general, this algorithm is used when X is described by primitive relations.

Coordinates

The algorithm `Coordinates[X, B]` gives the coordinates of generators of the fan of X with respect to a fixed basis B of the lattice N.

The input are:

1. the variety;
2. the basis of the lattice.

The output is a list of elements of type:

$$\texttt{x[i] -> \{x[i]}_1\texttt{,...,x[i]}_n\texttt{\}}$$

Example A.13.5. Let F_2 be the Hirzebruch surface $\mathbb{F}_2$ as in Example A.12.1:

```
In[1]:= base1 = Basis[F2]

Out[1]:= {x[1], x[3]}

In[2]:= Coordinates[F2, base1]

Out[2]:={x[1] -> {1, 0}, x[2] -> {-1, 0}, x[3] -> {0, 1}, x[4] ->
{2, -1}}
```

In general, we use this algorithm to compute coordinates of generators of the fan of the variety when the variety is described by primitive relations.

A.13.2 Contractibility

Contractibility test

The algorithm `ContractibleQ[X, rel]` says if `rel` is a contractible primitive relation.

The input are:

1. the variety described by primitive relations;
2. the primitive relation (it can be represented either by a polynomial or by a list list of type (A.2)).

The output is: `True` if the primitive relation is contractible, `False` otherwise.

The following example shows how the algorithm works:

Example A.13.6. Let T be the Oda 3-fold as in Example A.4.1.

```
In[1]:= ContractibleQ[T, {{x[5], x[6], x[7]}, {{x[4], 2}}}]
```

```
Out[1]:= True

In[2]:= ContractibleQ[T, x[1] + x[4] - x[7]]

Out[2]:= False
```

Set of contractible primitive relations as list of polynomials

The algorithm `ContractibleRel[X]` gives the list of all contractible primitive relations of X.

The input is the variety X described by the list of primitive relations.

The output is the set of all contractible primitive relations. In the list every primitive relation has the form of a linear polynomial in the generators.

Example A.13.7. Let T be the Oda 3-fold as in Example A.4.1.

```
In[1]:= ContractibleRel[T]

Out[1]:= {x[1] - x[2] + x[5] - x[7], x[2] - x[3] - x[5] + x[6],
-x[1] + x[3] - x[6] + x[7], -2 x[4] + x[5] + x[6] + x[7]}
```

Contractible primitive relations as list of lists

The algorithm `ListContractibleRel[X]` gives the list of all contractible primitive relations of variety X.

The input is the variety described by primitive relations.

The output is the list of all contractible primitive relation. In the list every primitive relation is represented by a list of type (A.2).

Example A.13.8. Let T be the Oda 3-fold, as in Example A.4.1.

```
In[1]:= ListContractibleRel[T]

Out[1]:= {{{x[1], x[5]}, {{x[2], 1}, {x[7], 1}}}, {{x[2], x[6]},
{{x[3], 1}, {x[5], 1}}}, {{x[3], x[7]}, {{x[1], 1}, {x[6], 1}}},
{{x[5], x[6], x[7]}, {{x[4], 2}}}}
```

A.13.3 Primitive relations

Computing primitive collections

Let X be a variety described by the data of generators and maximal cones. The algorithm `PrimColl[X]` allows to have the list of primitive collections of X.

The input is the variety X described by the data of generators and maximal cones.

The output is the list of primitive collection of X.

Example A.13.9. Let T be the Oda 3-fold as in Example A.5.2.

```
In[1]:= PrimColl[T]

Out[1]:= {{x[1], x[4]}, {x[1], x[5]}, {x[2], x[4]}, {x[2], x[6]},
{x[3], x[4]}, {x[3], x[7]}, {x[5], x[6], x[7]}}
```

Computing primitive relations

Let X be a variety described by the maximal cones of its fan and by generators. `PrimRel[X, coll]` gives the list of all primitive relations of X.

The input is the variety X described by the list of maximal cones.

The output is the list of primitive relations. Every primitive relation is represented by a list of type (A.2).

Example A.13.10. Let T be the Oda 3-fold as in Example A.5.2.

```
In[1]:= PrimRel[T, PrimColl[T]]

Out[1]:= {{{x[1], x[4]}, {{x[7], 1}}}, {{x[1], x[5]}, {{x[2], 1},
{x[7], 1}}}, {{x[2], x[4]}, {{x[5], 1}}}, {{x[2], x[6]}, {{x[3],
1}, {x[5], 1}}}, {{x[3], x[4]}, {{x[6], 1}}}, {{x[3], x[7]},
{{x[1], 1}, {x[6], 1}}}, {{x[5], x[6], x[7]}, {{x[4], 2}}}}
```

Set of primitive relations as list of polynomial

The algorithm gives a list of all primitive relations.

The input is the variety X described by primitive relations.

The output is the list of primitive relations of X represented by linear polynomials whose variables are generators of the fan of X.

Example A.13.11. Let T be the Oda 3-fold as in Example A.4.1:

```
In[1]:= ListRelCycles[T]

Out[1]:= {x[1] + x[4] - x[7], x[1] - x[2] + x[5] - x[7], x[2] +
x[4] - x[5], x[2] - x[3] - x[5] + x[6], x[3] + x[4] - x[6], - x[1]
+ x[3]  - x[6] + x[7],  -2 x[4] + x[5] + x[6] + x[7]}
```

Degree of a primitive relation

The algorithm `DegRel[rel]` gives the degree of a primitive relation.

The input is the primitive relation. It can have the form either of a linear polynomial or of a list.

The output is a number and it represents the degree of the primitive relation.

Example A.13.12. Let T be the Oda 3-fold as in Example A.4.1:

```
In[1]:= DegRel[{{{x[1], x[4]}, {{x[7], 1}}}]

Out[1]:= 1

In[2]:= DegRel[x[1] - x[2] + x[5] - x[7]]

Out[2]:= 0
```

A.13.4 Varieties

Describing a variety using primitive relations

Let X be a variety described by the data of generators and maximal cones of its fan. The algorithm `Varprimrel[X]` adds the list PR to the list $\{n, t, \mathrm{Null}, \mathrm{G}, \mathrm{MC}\}$.

The input is the variety described by generators and maximal cones of its fan.

The output is the variety X described by all possible descriptions.

Example A.13.13. Let T be the Oda 3-fold as in Example A.5.2.

```
In[1]:=  Varprimrel[T]

Out[1]:= {3, 7, {{{x[1], x[4]}, {{x[7], 1}}}, {{x[1], x[5]},
{{x[2], 1}, {x[7], 1}}}, {{x[2], x[4]}, {{x[5], 1}}}, {{x[2],
x[6]}, {{x[3], 1}, {x[5], 1}}}, {{x[3], x[4]}, {{x[6], 1}}},
{{x[3], x[7]}, {{x[1], 1}, {x[6], 1}}}, {{x[5], x[6], x[7]},
{{x[4], 2}}}}, {x[1] -> {1, 0, 0}, x[2] -> {0, 1, 0}, x[3] -> {0,
0, 1}, x[4] -> {-1, -1, -1}, x[5] -> {-1, 0, -1}, x[6] -> {-1, -1,
0}, x[7] -> {0, -1, -1}}, {{x[1], x[2], x[3]}, {x[1], x[2], x[7]},
{x[1], x[3], x[6]}, {x[1], x[6], x[7]}, {x[2], x[3], x[5]}, {x[2],
x[5], x[7]}, {x[3], x[5], x[6]}, {x[4], x[5], x[6]}, {x[4], x[5],
x[7]}, {x[4], x[6], x[7]}}}
```

We observe that the algorithm computes the primitive collections and relations using the algorithms `PrimColl[X]` and `PrimRel[X, coll]` respectively.

Describing a variety using maximal cones

Let X be a variety described by primitive relations. The algorithm `Varcones[X]` adds the lists G and MC to the list $\{n, t, \mathrm{PR}, \mathrm{Null}, \mathrm{Null}\}$.

The input is the variety described by primitive relations.

The output is the variety X described by the list $\{n, t, \mathrm{PR}, \mathrm{G}, \mathrm{MC}\}$.

Example A.13.14. Let T be the Oda 3-fold as in Example A.4.1:

```
In[1]:=  Varcones[T]

Out[1]:= {3, 7, {{{x[1], x[4]}, {{x[7], 1}}}, {{x[1], x[5]},
{{x[2], 1}, {x[7], 1}}}, {{x[2], x[4]}, {{x[5], 1}}}, {{x[2],
x[6]}, {{x[3], 1}, {x[5], 1}}}, {{x[3], x[4]}, {{x[6], 1}}},
{{x[3], x[7]}, {{x[1], 1}, {x[6], 1}}}, {{x[5], x[6], x[7]},
{{x[4], 2}}}}, {x[1] -> {1, 0, 0}, x[2] -> {0, 1, 0}, x[3] -> {0,
0, 1}, x[4] -> {-1, -1, -1}, x[5] -> {-1, 0, -1}, x[6] -> {-1, -1,
0}, x[7] -> {0, -1, -1}}, {{x[1], x[2], x[3]}, {x[1], x[2], x[7]},
{x[1], x[3], x[6]}, {x[1], x[6], x[7]}, {x[2], x[3], x[5]}, {x[2],
x[5], x[7]}, {x[3], x[5], x[6]}, {x[4], x[5], x[6]}, {x[4], x[5],
x[7]}, {x[4], x[6], x[7]}}}
```

A.13.5 Smooth test

`SmoothQ[X]` says if the variety X is smooth or not.

The input is the variety X.

The output is a string: `True` if the variety is smooth and `False` otherwise.

Example A.13.15. Let T be the Oda 3-fold as in Example A.5.2.

```
In[1]:= SmoothQ[T]

Out[1]:= True
```

A.13.6 Curves

List of polynomial representing all different numerical classes of invariant curves

The algorithm `AllInvCurves[X]` gives the list of all relations associated to all different numerical classes of invariant curves in X.

The input of algorithm is the variety X.

The output is a list of polynomials in the generators of the fan of X.

Example A.13.16. Let T be the Oda 3-fold as in Example A.5.2.

```
In[1]:=  AllInvCurves[T]

Out[1]:= {x[2] + x[4] - x[5], x[1] + x[3] + x[5], x[3] + x[4] -
x[6], x[1] + x[2] + x[6], x[2] - x[3] - x[5] + x[6], x[1] + x[4] -
x[7], x[1] - x[2] + x[5] - x[7], x[2] + x[3] + x[7], -x[1] + x[3]
- x[6] + x[7], -2 x[4] + x[5] + x[6] + x[7]}
```

The maximal cones associated to an invariant curves

Given the variety X, an $(n-1)$-dimensional cone τ in the fan determines an invariant curve C, where $C = V(\tau)$ in X.

The input are:

1. the variety;
2. the set of generators of the $(n-1)$-cone τ.

The output of the algorithm `MaximalCones[X, cone]` is the list of the two maximal cones σ, σ' such that $\tau = \sigma \cap \sigma'$. They are given by the list of their generators.

The next example shows how the algorithm works:

Example A.13.17. Let T the Oda 3-fold as in Example A.4.1.

```
In[1]:= MaximalCones[T, {x[1], x[2]}]

Out[1]:= {{x[1], x[2], x[3]}, {x[1], x[2], x[7]}}
```

A.13.7 Extremal classes

List of coordinates of numerical classes of invariant curves and of coordinates of contractible primitive relations in $\mathcal{N}_1(X)_{\mathbb{Q}}$

Let $\mathcal{I}$ be the set of all different numerical classes of all invariant curves of the variety X. It contains a basis B of $\mathcal{N}_1(X)_{\mathbb{Q}}$.

If X is also projective, we consider the set $\mathcal{C}$ of all contractible primitive relations, because it contains a basis of $\mathcal{N}_1(X)_{\mathbb{Q}}$.

The algorithm determines the basis B and computes the coordinates of every element in $\mathcal{I}$, (respectively in $\mathcal{C}$), with respect to B.

The input are:

1. the variety X;
2. either the set $\mathcal{I}$ or the set $\mathcal{C}$, whose elements are polynomial in the generators of the fan.

The output is a list of all elements of $\mathcal{I}$, (respectively of $\mathcal{C}$), with their coordinates. If you have the element $\mathtt{a}_{\mathtt{i}_1}\mathtt{x[i}_1\mathtt{]} + \cdots + \mathtt{a}_{\mathtt{i}_k}\mathtt{x[i}_k\mathtt{]}$ in $\mathcal{I}$, (respectively in $\mathcal{C}$) and $\{\mathtt{b}_1, \ldots, \mathtt{b}_\rho\}$ are its coordinates, then, in the list you find:

$$\mathtt{a}_{\mathtt{i}_1}\mathtt{x[i}_1\mathtt{]} + \cdots + \mathtt{a}_{\mathtt{i}_k}\mathtt{x[i}_k\mathtt{]} \;\texttt{->}\; \mathtt{b}_1, \ldots, \mathtt{b}_\rho$$

Example A.13.18. Let W be the variety of Example A.6.1:

```
In[1]:= EsprVett[W, ContractibleRel[W]]

Out[1]:={x[1] + x[5] - x[8] -> {1, 0, 0, 0, 0}, x[2] - x[3] - x[5]
+ x[6] -> {-1, 1, -1, 0, 0}, x[2] + x[7] - x[8] -> {0, 1, 0, 0,
0}, - x[1] + x[3] - x[6] + x[7] -> {0, 0, 1, 0, 0}, x[4] - x[5] -
x[7] + x[8]" -> {0, 0, 0, 1, 0}, -2 x[4] + x[5] + x[6] + x[7] ->
{0, 0, 0, 0, 1}}

In[2]:= EsprVett[W, AllInvCurves[W]]

Out[2]:= {x[1] + x[3] + x[5] -> {0, 1, 1, 2, 2}, x[3] + x[4] -
x[6] -> {0, 1, 0, 1, 1}, x[1] + x[2] + x[6] -> {1, 1, 1, 2, 2},
x[2] - x[3] - x[5] + x[6] -> {1, 0, 0, 0, 0}, x[1] + x[4] - x[7]
-> {0, 0, 0, 1, 1}, -x[1] + x[3] - x[6] + x[7] -> {0, 1, 0, 0, 0},
-2 x[4] + x[5] + x[6] + x[7] -> {0, 0, 1, 0, 0}, x[1] + x[5] -
x[8] -> {0, 0, 0, 1, 0}, x[2] + x[7] - x[8] -> {1, 1, 0, 1, 0},
x[3] + x[8] -> {0, 1, 1, 1, 2}, x[4] - x[5] - x[7] + x[8] -> {0,
0, 0, 0, 1}, x[1] + x[6] - x[7] + x[8] -> {0, 0, 1, 1, 2}}
```

The list of extremal and non-extremal classes

Let X be projective variety. Suppose that you want to describe the Mori cone $\mathrm{NE}(X)$. The algorithm `ExtremalClasses[X]` is a "list-version" of the algorithm `TabExtremalClasses[X]`.

The input of algorithm is the variety X.

The output is a list of two sets:

1. in the first, you find all extremal classes;
2. in the second, there are the non-extremal classes.

Every element is given together with its coordinates with respect to a basis B of $\mathcal{N}_1(X)_{\mathbb{Q}}$.

The following example shows how the algorithm works.

Example A.13.19. Let W be the variety as in Example A.6.1:

```
In[1]:= W = {3, 8, {{{x[1], x[4]}, {{x[7], 1}}}, {{x[1], x[5]},
{{x[8], 1}}}, {{x[2], x[4]}, {{x[5], 1}}}, {{x[2], x[6]}, {{x[3],
```

```
1}, {x[5], 1}}}, {{x[2], x[7]}, {{x[8], 1}}}, {{x[3], x[4]},
{{x[6], 1}}}, {{x[3], x[7]}, {{x[1], 1}, {x[6], 1}}}, {{x[3],
x[8]}, {{0, 0}}}, {{x[4], x[8]}, {{x[5], 1}, {x[7], 1}}}, {{x[6],
x[8]}, {{x[4], 1}}}, {{x[5], x[6], x[7]}, {{x[4], 2}}}}, Null,
Null};

In[2]:= ExtremalClasses[W]

Out[2]:= {{x[2] - x[3] - x[5] + x[6] -> {-1, 1, -1, 0, 0}, x[1] +
x[5] - x[8] -> {1, 0, 0, 0, 0}, -x[1] + x[3] - x[6] + x[7] -> {0,
0, 1, 0, 0}, x[4] - x[5] - x[7] + x[8] -> {0, 0, 0, 1, 0}, -2 x[4]
+ x[5] + x[6] + x[7] -> {0, 0, 0, 0, 1}}, {x[2] + x[7] - x[8] ->
{0, 1, 0, 0, 0}}}
```

A.14 Error messages

Some algorithms also answer with a message of error. We present the list of error messages of every algorithm and explain the reasons of this answer.

They are collected following the alphabetically order of the name of algorithms.

1. `AddInput[X]`:

   `Error: it is not possible to compute coordinates of generators.`
   The message is visualized if the system given by primitive relations of the variety X does not allow to determine coordinates of generators of the fan.

2. `AllInvCurves[X]`:

   `Error: it is not possible to compute the relation associated to a curve.`
   It appears if it is not possible to know the relation associated to an invariant curve because the system given by primitive relations of the variety X does not allow to compute the coordinates of every generator of the fan.

3. `AnticanonicalNefQ[X]`:

   `Error: it is not possible to compute the relation associated to a curve.`
   It is visualized if it is not possible to know the relation associated to an invariant curve because the system given by primitive relations of the variety X does not allow to compute the coordinates of every generator of the fan.

4. `BasisQ[X,set]`:

   `Error: it is not possible to compute coordinates of generators.`
   The message is visualized if the system given by primitive relations of the variety does not allow to determine the coordinates of the generators of the fan.

5. `BasisCoord[X]`:

   `Error: it is not possible to compute coordinates of generators.`

   It appears if the system given by primitive relations of the variety does not allow to determine the coordinates of the generators of the fan.

6. `Blowdown[X,x[h]]`:

   (a) `Error: there are not contractible primitive relations.`

   This message is visualized if the variety has not a contractible primitive relation of type `x[1] +` $\cdots$ `+ x[k] - x[h] = 0` for the generator `x[h]`.

   (b) `Error: the contraction of the class is not a smooth blow-down.`

   This string appears if there is only one contractible primitive relation $r(P)$ with `x[h]`$\in \sigma_P$ but not having the form `x[1] +` $\cdots$ `+ x[k] - x[h] = 0`.

7. `Blowup[X,cone]`:

   (a) `Error: the second input does not span a cone in the fan.`

   It is visualized if the second input does not generate a cone in the fan of the variety.

   (b) `Error: it is not possible to compute coordinates of generators.`

   It appears if the system given by primitive relations of the variety X does not allow to compute the coordinates of the generators of the fan.

8. `Cones[X,k]`:

   `Error: the second input is bigger than the dimension of the variety.`

   This message marks out that the choice for the dimension of the cones is wrong: it is a positive integer k such that $k \leq n$.

9. `ContractibleQ[X, rel]`:

   `Error: the variety is described by maximal cones.`

   This message appears if the variety is described by the data of generators and maximal cones.

10. `ContractibleRel[X]`:

    `Error: the variety is described by maximal cones.`

    The string is visualized if the variety is described by the data of generators and maximal cones.

11. `CoordCurvesClasses`:

    `Error: it is not possible to compute the relation associated to a curve.`

    It appears if it is not possible to know the relation associated to an invariant curve because the system given by primitive relations of the variety X does not allow to compute the coordinates of every generator of the fan.

12. `Coordinates[X]`:

    `Error: it is not possible to compute the coordinates of generators`.

    The message appears when the system given by primitive relations of the variety X does not allow to determine coordinates of generators of the fan.

13. `Curve[X, cone]`:

    (a) `Error: the set cone does not span a cone in the fan`.

    It marks out that the second input does not generate a cone in the fan of the variety.

    (b) `Error: it is not possible to compute coordinates of generators`.

    It is visualized when the system given by primitive relations of X does not allow to determine coordinates of generators of the fan of the variety.

14. `DelPezzo[n]`:

    `Error: the dimension is odd`.

    This message appears because the dimension of a Del Pezzo variety is even.

15. `FanoQ[X]`:

    `Error: it is not possible to compute the relation associated to a curve`.

    It is visualized if it is not possible to know the relation associated to an invariant curve because the system given by primitive relations of X does not allow to compute the coordinates of every generator of the fan.

16. `ListContractibleRel[X]`:

    `Error: the variety is described by maximal cones`.

    The string is visualized if the variety is described by the data of generators and maximal cones.

17. `ListRelCycles[X]`:

    `Error: the variety is described by maximal cones`.

    This message is visualized if the variety is described by the data of generators and maximal cones.

18. `Prod[X]`

    `Error: it is not possible to compute the product`.

    It appears if the system given by primitive relations of the variety does not allow to determine coordinates of generators of the fan.

19. `ProjQ[X]`:

    `Error: it is not possible to compute the relation associated to a curve`.

    It is visualized if it is not possible to know the relation associated to an invariant curve because the system given by primitive relations of X does not allow to compute the coordinates of every generator of the fan.

20. `PseudoDelPezzo[n]`:

    `Error: the dimension is odd`.

    The string appears because the dimension of a pseudo Del Pezzo variety is even.

21. `TabContrRel[X]`:

    `Error: the variety is described by maximal cones`.

    This message is visualized if the variety is described by the data of generators and maximal cones.

22. `TabInvCurves[X]`:

    `Error: it is not possible to compute the relation associated to a curve`.

    It appears if it is not possible to know the relation associated to an invariant curve because the system given by primitive relations of X does not allow to compute the coordinates of every generator of the fan.

23. `TabVar[X]`:

    `Error: the variety is described by maximal cones`.

    It is visualized if the variety is described by the data of generators and maximal cones.

24. `Varcones[X]`:

    `Error: it is not possible to compute coordinates of generators`.

    It marks out that the system given by primitive relations of the variety does not allow to determine the coordinates of the generators of the fan.

A.15 Help messages

Every algorithm is equipped of an help, in this way you are made easier to use them.

If you want to know what a function computes, it is enough that you use the following command:

```
In[1]:= ? namefunction
```

where `namefunction` is the name of the command.

For example, suppose that you want to know what the command `ProjSpace` computes then you write:

```
In[1]:= ? ProjSpace
```

and *Mathematica* answers with the string:

```
ProjSpace[n] constructs the projective space of dimension n.
```

Notice that the string says you also the input of the function.

If you want to look at the text of the algorithm `namefunction` you have to use the following command:

```
In[1]:= ?? namefunction
```

Suppose that you want to know the text of command `ProjSpace` then you write

```
In[1]:= ?? ProjSpace
```

Mathematica answers with:

```
ProjSpace[n_]:=Block[{coll, generat, sottoins, i, base, vett,
coord,risultato}, coll = Table[x[i], {i, 1, n + 1}]; generat
=Table[x[i], {i, n + 1}]; <<"DiscreteMath`Combinatorica`";
sottoins = KSubsets[generat, n]; base = IdentityMatrix[n]; vett =
Table[-1, {i, n}]; coord = Table[x[i] -> base[[i]], {i,n}]; coord
= AppendTo[coord, x[n + 1] -> vett]; risultato = {n, n + 1,{{coll,
{{0, 0}}}}, coord, sottoins}]
```

You find the following help messages (they are in the alphabetically order):

1. `AddInput`

   `AddInput[X] adds information to the list describing X replacing the Null entries with the corresponding missing list.`

2. `AllInvCurves`

   `AllInvCurves[X] computes all relations associated to all different numerical classes of invariant curves in X.`

3. `AnticanonicalNefQ`

   `AnticanonicalNefQ[X] says if the anticanonical bundle is nef or not.`

4. `Basis`

   `Basis[X] computes a basis of lattice N contained in the set of generators of the fan of X.`

5. `BasisCoord`

   `BasisCoord[X] computes a basis B of lattice N contained in the set of generators of the fan of X and the coordinates of all generators with respect to B.`

6. BasisQ

   BasisQ[X,set] verifies if set is a basis of lattice N.

7. Blowdown

   Blowdown[X,z] computes the blowdown of X with respect to a contractible relation of type x[1]+···+x[k]-z=0.

8. Blowup

   Blowup[X,set] computes the blowup of X along the invariant subvariety corresponding to the cone generated by set.

9. ConeQ

   ConeQ[X,set] verifies if set spans a cone in the fan of X.

10. Cones

    Cones[X,k] computes the k-dimensional cones in the fan of X.

11. ContractibleQ

    ContractibleQ[X,rel] says if rel is a contractible primitive relation.

12. ContractibleRel

    ContractibleRel[X] computes all contractible primitive relations as linear polynomials in the generators of the fan.

13. Coordinates

    Coordinates[X,B] gives the coordinates of all generators of the fan of X with respect to the basis B.

14. Curve

    Curve[X,set] computes the relation associated the numerical class of the invariant curve corresponding to the cone generated by set.

15. CoordCurvesClasses

    CoordCurvesClasses[X] computes a basis B in the set of numerical classes of curves and the coordinates of all numerical classes of curves with respect to B.

16. CoordContrClasses

    CoordContrClasses[X] computes a basis B in the set of contractible classes of the projective variety X and the coordinates of all contractible classes with respect to B.

17. DelPezzo

    DelPezzo[n] constructs the Del Pezzo variety of dimension n.

18. DegRel

    DegRel[rel] computes the degree of the primitive relation rel.

19. EsprVett

    EsprVett[X,set] computes a basis B and the coordinates with respect to B of all classes in set. set is either the set of all numerical classes of invariant curves or the set of all contractible classes of X. The answer is a list.

20. ExtremalClasses

    ExtremalClasses[X] gives the lists of all extremal and non-extremal classes of the projective variety X.

21. FanoQ

    FanoQ[X] says if X is Fano or not.

22. Gen

    Gen[X] computes the set of generators of the fan of X.

23. IsomQ

    IsomQ[X1,X2] says if X1 and X2 are isomorphic.

24. `ListContractibleRel`

`ListContractibleRel[X]` computes all contractible primitive relations, every relation has the form of a list.

25. `ListRelCycles`

`ListRelCycles[X]` gives all primitive relations as polynomials in the generators of the fan.

26. `MaximalCones`

`MaximalCones[X,set]` computes the two maximal cones whose intersection is the cone generated by set.

27. `NewVariety`

`NewVariety` allows to introduce a smooth complete toric variety.

28. `PrimColl`

`PrimColl[X]` computes all primitive collections of X.

29. `PrimRel`

`PrimRel[X,coll]` computes the primitive relations of all primitive collections contained in the set coll.

30. `Prod`

`Prod[X1,X2]` computes the product of varieties X1 and X2.

31. `ProjSpace`

`ProjSpace[n]` constructs the projective space of dimension n.

32. `ProjQ`

`ProjQ[X]` verifies if X is projective or not.

33. `PseudoDelPezzo`

`PseudoDelPezzo[n]` constructs the pseudo Del Pezzo variety of dimension n.

34. `SmoothQ`

`SmoothQ[X] says if X is smooth.`

35. `TabContrRel`

`TabContrRel[X] constructs the table of all contractible primitive relations of X.`

36. `TabInvCurves`

`TabInvCurves[X] gives the table of all relations associated to the numerical classes of the invariant curves together with the set of generators of the corresponding cones in the fan.`

37. `TabExtremalClasses`

`TabExtremalClasses[X] constructs the table of all contractible primitive relations together their coordinates and specifies that a class is extremal adding the letter E.`

38. `TabVar`

`TabVar[X] constructs the table of all primitive relations of X.`

39. `Varcones`

`Varcones[X] adds the list of generators and of maximal cones to the list which describes X.`

40. `Varprimrel`

`Varprimrel[X] adds the list of all primitive relations to the list which describes X.`

Bibliography

[AC] Farid Ajili, Evelyne Contejean, *Complete solving of linear and Diophantine equational and disequational systems without adding variables*, Technical Report 0175, INIRIA, June 1995.

[AL] William W. Adams, Philippe Loustaunau, *An Introduction to Gröbner Bases*, Graduate Sudies in Mathematics, vol. 3, 1994.

[AMR] Dan Abramovich, Kenji Matsuki, Suliman Rashid, *A note on the factorization theorem of toric birational maps after Morelli and its toroidal extension*, Tohoku Math. J. (2) 51(1999), no. 4, pages 489-537.

[Ba1] V. V. Batyrev, *Toroidal Fano 3-folds*, Mathematical of USSR Izvestiya, vol. 19, pagg. 13-25, 1982.

[Ba2] V. V. Batyrev, *On the classification of smooth projective toric varieties*, Tohoku Mathematical Journal, vol. 43, pages 569-585, 1991.

[Ba3] V. V. Batyrev, *Dual polyedra and mirror symmetry for Calaby-Yau hypersurfaces in toric varieties*, J. Algebraic Geom. 3, pages 493-535, 1994.

[Ba4] V. V. Batyrev, *On the classification of toric Fano 4-folds*, Journal of Mathematical Sciences (New York), vol. 94, pages 1021-1050, 1999.

[Bo] Laurent Bonavero, *Sur de Variétés Toriques non Projectives*, Bull. Soc. Math. France, vol. 128, pages 407-431, 2000.

[C1] Cinzia Casagrande, *Toric Varieties: Equivariant Contractions and Fano Varieties*, Tesi di Dottorato, 2001.

[C2] Cinzia Casagrande, *Contractible classes in toric varieties*, Mathematische Zeitschrift, 243, vol. 1, pages 99-126, 2003.

[CJR] Cinzia Casagrande, Priska Janke, Ivo Radloff, *On the Picard number of almost Fano threefolds with pseudo-index > 1*, Preprint math. AG/0603090.

[CD] Evelyne Contejean, Hervé Devie, *An efficient incremental algorithm for solving systems of linear Diophantine equations*, Inform. and Comput., 113(1994), no. 1, pages 143-172.

[CLO1] David A. Cox, John Little, Donal O'Shea, *Ideal, Varieties and Algorithms - An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Second Edition, Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1997.

[CLO2] David A. Cox, John Little, Dona O'Shea, *Using Algebraic Geometry*, second edition, Graduate Texts in Mathematics, vol. 185, Springer, New York, 2005.

[CP] A. H. Clifford, G. B. Preston, *The Algebraic Theory of Semigroups*, Amer. Math. Society, vol. 2, 1967.

[dps] *Diophantine Polynomial Systems* - available on the web page http:// documents.wolfram.com / mathematica / functions / AdvancedDocumentationDiophantinePolynomialSystems

[E1] Günter Ewald, *On the classification of toric Fano varieties*, Discrete Computational Geometry, n. 3, pages 49-54, 1988.

[E2] Günter Ewald, *Combinatorial Convexity and Algebraic Geometry*, Graduate Texts in Mathematics, Springer - Verlag, vol. 168, 1996.

[F] William Fulton, *Introduction to Toric Varieties*, no 131 in Annales of Mathematics Studies, Princeton University Press, 1993.

[FP] Osamu Fujino, Sam Payne, *Smooth complete toric threefolds with no nontrivial nef line bundles*, Proc. Japan Acad., vol. 81, Ser A, pages 174-179, 2005.

[FS] Osamu Fujino, Hiroshi Sato, *Introduction to the Toric Mori Theory*, Michigan Math. J., 52(2004), no. 3, Preprint math. pages 649-665.

[Kl] Steven L. Kleiman, *Toward a numerical thery of ampleness*, Annales of Mathematics, n. 84, pages 293-344, 1966.

[Kr] Andrew Kresch, *Gromov-Witten Invariants of a Class of Toric Varieties*, Michigan Math. J. 48(2000), pages 369-391.

[KlSt] Peter Kleinschmidt, Bernd Sturmfels, *Smooth toric varieties with small Picard number are projective*, Topology 30(1991), no. 2, pages 289-299.

[KrSk] Maximilian Kreuzer, Harald Skarke, *Classification of Reflexive Polyedra in Three Dimension*, Adv. Theor. Math. Phys., no. 2, pages 853-871, 1998.

[L] Serge Lang, *Linear Algebra*, Undergraduate Texts in Mathematics edited by F.W. Gehring, P.R. Halmos, Springer, New York, 1989.

[list] *List of all 4319 reflexive polyhedra in 3 dimensions* - available on the web page http:// hep.itp.tuwien.ac.at / kreuzer / CY /

[Ma] Kenji Matsuki, *Correction: A note on the factorization theorem of toric birational maps after Morelli and its toroidal extension* (by D. Abramovich, Matsuki and S. Rashid. Tohoku Math. J. (2) 52 (2000), no. 4,629-631), Tohoku Math. J. (2) 51 (1999), no. 4, pages 489-537.

[Mo] Robert Morelli, *The birational geometry of toric varieties*, J. Algebraic Geom., 5 (1996), no. 4, pages 751-782.

[N] Benjamin Nill, *Gorestein toric Fano varieties*, Manuscripta Math., 116, pages 183-210, 2005.

[O1] Tadao Oda, *Torus Embeddings and Applications (Based on joint work with Katsuya Miyake)* , Published for the Tata Institute of Fundamental Research Bombay, Tata Institute of Fundamental Research, Lectures on Mathematics and Physics, Mathematics, Springer - Verlag, 1978.

[O2] Tadao Oda, *Convex Bodies and Algebraic Geometry - An Intrduction to the Theory of Toric Varieties - An Introduction to the Theory of Toric Varieties*, Springer - Verlag, 1988.

[R] Miles Reid, *Decomposition of toric morphisms*, Arithmetics and Geometry, dedicated to I. R. Shafarevich, Geometry, Prog. Math., vol. 2, pages 395-418, 1988.

[Sa] Hiroshi Sato, *Toward the Classification of Higher Dimension of Toric Fano Varieties*, Tohoku Mathematical Journal, vol. 52, pages 383-413, 2000.

[Se] Edoardo Sernesi, *Geometria 1*, Bollati Boringhieri, Ristampa 1994.

[VK] V. E. Voskresenskiĭ, Alexander Klyachko, *Toroidal Fano varieties and roots systems*, Mathematics of USSR Izvestiya, n. 24, pages 221-224, 1985.

[Wi] Jarosław A. Wiśniewski, *Toric Mori Theory and Fano Manifolds*, Seminaires et Congrès, 6, pages 249-272, 2002.

[Wo] Stephen Wolfram, *The Mathematica Book*, 5-th ed, (Wolfram Media, 2003).

[WW] Keiichi Watanabe, Masayuki Watanabe, *The classification of Fano 3-folds with torus embeddings*, Tokyo Journal of Mathematics, 5, pages 37-48, 1982.

Acknowledgments

I would like thank Lucia Caporaso for following me in the first year of my Ph.D. and for the possibility to work with Cinzia Casagrande. I thank Cinzia Casagrande for her guidance and patience throughout the preparation of this thesis.

I would like to thanks the referees for their careful reading and the useful comments.

I acknowledge the Dipartimento di Matematica “Roma Tre” at Rome for the support given during the preparation of the thesis. I would like thank the Dipartimento di Matematica of the Univeristy of Pisa for the kind hospitality.

I’m grateful to the Ph.D. students who contributed to create a nice atmosphere at the Dipartimento di Mathematica in Rome. In particular, I thank my colleague and friend Andrea Susa for his company and kindness.

I wish to thank my family for their encouragement and support. A big thanks goes to all my friends: their friendship is a very special support for me.