Oriented Cohomology Rings of the Semisimple Linear Algebraic Groups of Ranks 1 and 2

Raj Gandhi

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science Mathematics and Statistics[1]

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

[1]The M.Sc. program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Abstract

In this thesis, we compute minimal presentations in terms of generators and relations for the oriented cohomology rings of several semisimple linear algebraic groups of ranks 1 and 2 over algebraically closed fields of characteristic 0. The main tools we use in this thesis are the combinatorics of Coxeter groups and formal group laws, and recent results of Calmès, Gille, Petrov, Zainoulline, and Zhong, which relate the oriented cohomology rings of flag varieties and semisimple linear algebraic groups to the dual of the formal affine Demazure algebra.

Dedications

To my family.

Acknowledgements

I would first and foremost like to thank my Supervisors Alistair Savage and Kirill Zaynullin for sharing with me their enthusiasm for mathematics and for the ideas they have brought forth towards this thesis. This work would not have been possible without their guidance. I would also like to thank my Thesis Examiners Colin Ingalls and Erhard Neher for providing feedback that greatly improved the quality of this thesis. I acknowledge the financial support provided to me by the University of Ottawa, the Natural Sciences and Engineering Research Council of Canada, and an Ontario Graduate Scholarship during my Master's studies. Finally, I would like to thank my family, who have always supported me.

Contents

General notation

$\mathbb{Z}$	Integers
$\mathbb{Q}$	Rational numbers
$\mathbb{R}$	Real numbers
$\mathbb{C}$	Complex numbers
$\|X\|$	Cardinality of the set X

Chapter 1

Introduction

1.1 Background

Computing cohomology rings of connected compact Lie groups and Chow rings of connected reductive algebraic groups was of great interest in the 1900s. Cartan, Brauer, Pontryagin, and others computed the real cohomology of the classical compact Lie groups between 1929 and 1935. By 1949, Hopf, Samelson, Yen, and others had computed the real cohomology of the exceptional Lie groups. The torsion in the integral cohomology of the classical compact Lie groups was computed between 1953 and 1965 by Borel, Baum-Browder, Miller, and others. The introduction of Kac's paper [25] gives a detailed history of these computations. In his 1985 paper [25], Kac provided an algorithm to compute the cohomology ring of K with coefficients in a finite field, $\mathrm{H}^*(K;\mathbb{F}_p) = \mathrm{H}^*(K) \otimes_{\mathbb{Z}} \mathbb{F}_p$, when $p > 2$ is prime and K is any connected compact Lie group, or $p = 2$ and K is a simply-connected compact Lie group. If G is the complex reductive group corresponding to K, then, in [25], Kac simultaneously computes the Chow ring with coefficients in a finite field, $\mathrm{CH}^*(G;\mathbb{F}_p) = \mathrm{CH}^*(G)\otimes_{\mathbb{Z}}\mathbb{F}_p$.

The program of computing the integral cohomology rings of the compact connected Lie groups was not completed until recently. In 1991 [34], Pittie computed the integral cohomology of the Lie groups $\mathrm{SO}(n)$ and $\mathrm{Spin}(n)$. In 2015 [13], Duan-Zhao computed minimal presentations in terms of generators and relations for the integral cohomology rings of the five exceptional complex Lie groups of adjoint type, and, in the same year [12], Duan-Zhao computed the integral cohomology rings of the 1-connected Lie groups. Finally, in 2016, [11], Duan computed the integral cohomology rings of the remaining compact connected Lie groups.

Computing generalized cohomology rings for the connected compact Lie groups became of interest in the 1990s and continues to be of interest today. By the early 1990s, it was known that the K-theory of a simply connected Lie group is isomorphic to $\mathbb{Z}$ (see, for example, Levine [30]). In 2005 [44], Yagita computed the algebraic cobordism modulo a certain ideal of the simply-connected complex reductive groups

G. In addition, when G is simple, Yagita computed a minimal presentation for the algebraic cobordism ring of G modulo a certain ideal in terms of generators and relations.

1.2 New results and future directions

The goal of this thesis is to provide a presentation for the oriented cohomology rings of all semisimple connected linear algebraic groups over algebraically closed fields k of characteristic 0 (This includes the algebraic cobordism rings, without taking the quotient by any nonzero ideals). This is the content of Theorem 5.3.10 and Lemma 5.3.11, which are new results. As an application of this presentation, we compute *minimal* presentations in terms of generators and relations (i.e., presentations with the fewest number of generators possible and, given the number of generators, the fewest number of relations possible) for the oriented cohomology rings of several semisimple linear algebraic groups of ranks 1 and 2 in Section 5.4. These computations are also new and generalize the computations of all the authors mentioned in Section 1.1. In the appendices of this thesis, we have included the Python code that we used to compute these minimal presentations. It is very likely that this code can be adapted to compute minimal presentations in terms of generators and relations for the oriented cohomology rings of the semisimple linear algebraic groups of rank greater than 2.

The Python code included in the appendices is available online: [16]. The author expects to submit a paper containing most of the results in Chapter 5 in the more general setting of split semisimple linear algebraic groups over arbitrary fields shortly to a research journal.

1.3 Organization and content of this thesis

We will now discuss the content and organization of this thesis in detail. The first three chapters of this thesis develop the vocabulary and technical tools needed to understand the oriented cohomology rings of semisimple linear algebraic groups over algebraically closed fields of characteristic 0. These chapters are mainly expository and should be accessible to readers with a background in general topology at the level of Munkres [32] and abstract algebra at the level of Dummit-Foote [14]. We note that some results covered in the first three chapters appear to be well known to experts, but do not appear to be fully justified in the literature. We attempt to fill this gap in the literature by proving such results ourselves:

- In Section 2.3, we prove Theorem 2.3.7, Theorem 2.3.8, Proposition 2.3.9, and Proposition 2.3.10, which imply that, in a uniform space, all Cauchy nets converge if and only if all Cauchy filters converges. These four results appear to be well-known to experts. However, we have not been able to find proofs of these results in the literature.

- In Section 3.1, we compute the first six coefficients in the formal inverse of any one-dimensional commutative formal group law in terms of the coefficients of the formal group law. We have not been able to find these explicit formulas in the literature.

We will now discuss the organization of the first three chapters. In Chapter 2, we discuss basic category theory, algebras and coalgebras over commutative unital rings, and topological groups and rings. In Chapter 3, we discuss formal group laws, algebraic varieties and vector bundles, and oriented cohomology theories. In Chapter 4, we discuss linear algebraic groups. We begin by defining root systems, Dynkin diagrams, weight lattices, and root data. We then define the linear algebraic groups, and show how reductive connected linear algebraic groups can be classified in terms of root data. We also define flag varieties, and discuss line bundles on complete flag varieties.

We will now discuss the content of the fourth chapter. Let h^* be an oriented cohomology theory in the sense of Levine-Morel [31]. Examples of oriented cohomology theories include the Chow theory CH^*, Grothendieck's K^0 functor, and the universal oriented cohomology theory called algebraic cobordism Ω^*. Let G be a semisimple linear algebraic group over an algebraically closed field k of characteristic 0, and fix a Borel subgroup B of G, so that G/B is a complete flag variety. Let T be a maximal torus of G sitting in B, and let Λ be the character lattice of T. In [17, Theorem 5.1], Gille-Zainoulline derived a relationship between the oriented cohomology rings $\mathrm{h}^*(G)$ and $\mathrm{h}^*(G/B)$ through the first Chern class map $\mathrm{c}_1^{\mathrm{h}^*}$ when h^* satisfies certain conditions (see Section 5.3). The precise relationship is

$$\mathrm{h}^*(G) \simeq \mathrm{h}^*(G/B)/(\mathrm{c}_1^{\mathrm{h}^*}(\mathcal{L}(\lambda_1)), \ldots, \mathrm{c}_1^{\mathrm{h}^*}(\mathcal{L}(\lambda_n))),$$

where $\{\lambda_i\}$ is a $\mathbb{Z}$-basis of Λ, and $\mathcal{L}(\lambda_i)$ is the line bundle over G/B corrsponding to λ_i. In [7], Calmès-Petrov-Zainoulline constructed an algebraic model for $\mathrm{h}^*(G/B)$ using the *augmentation* of an algebra $\epsilon\mathcal{D}_F$ containing formal Demazure operators. The ideal $(\mathrm{c}_1^{\mathrm{h}^*}(\mathcal{L}(\lambda_1)), \ldots, \mathrm{c}_1^{\mathrm{h}^*}(\mathcal{L}(\lambda_n)))$ in $\mathrm{h}^*(G/B)$ corresponds to the image in the dual $(\epsilon\mathcal{D}_F)^*$ of an *algebraic* characteristic map. Using this information, we obtain an algebraic model for $\mathrm{h}^*(G)$ in Theorem 5.3.10. We then describe this algebraic model in terms of generators and relations in Lemma 5.3.11. With enough computing power, it is possible to write down the generators and relations in the ring $\mathrm{h}^*(G)$ explicitly, and, from these relations, one can compute a minimal presentation for $\mathrm{h}^*(G)$ in terms of generators and relations. In Section 5.4, we perform this calculation for several semisimple linear algebraic groups of ranks 1 and 2.

Now fix an adjoint semisimple linear algebraic group G of rank 2 over an algebraically closed field k of characteristic 0. There are three appendices in this thesis, which contain the Python code that we used to compute the relations in $\mathrm{h}^*(G)$. The code in Appendix A computes the first six coefficients in the formal inverse of any one-dimensional commutative formal group law. The code in Appendix B computes certain coproduct coefficients necessary to understand the relations in $\mathrm{h}^*(G)$. The code in Appendix C computes all the relations in $\mathrm{h}^*(G)$ using the output of the code in the first two appendices. We note that the output of the code in Appendix C does not give a minimal presentation for $\mathrm{h}^*(G)$ in terms of generators and relations. However, in Section 5.4, we use the output of the code in Appendix C to compute a minimal presentation for $\mathrm{h}^*(G)$ in terms of generators and relations by hand.

Chapter 2

Categories and commutative algebra

In this chapter, we discuss basic category theory, algebras and coalgebras, and topological groups and topological rings. In Section 2.1, we introduce basic notions in category theory. We use the language of category theory from time-to-time in this thesis. In Section 2.2, we discuss algebras and coalgebras, which will be particularly useful in Section 5.2. In Section 2.3, we discuss topological groups and topological rings. Our exposition of category theory can be found in most textbooks that cover category theory; see, for example, Leinster [29]. Our exposition of algebras and coalgebras closely follows the exposition in Kassel [26, Part 1, §III] and Underwood [42, Ch. 4]. Our exposition of topological group and topological rings closely follows Atiyah-Macdonald [1], Bourbaki [4] and [5], Husain [24], and Nagata [33].

2.1 Category theory

In this section, we introduce several basic notions in category theory. We discuss categories, functors, and natural transformations, and we give many examples of these concepts. We closely follow [29].

Definition 2.1.1. *A* category $\mathcal{C}$ *consists of the following data:*

- *A collection* $\mathrm{Ob}(\mathcal{C})$ *of* objects*.*
- *A collection* $\mathrm{Mor}(\mathcal{C})$ *of* morphisms*. Every morphism f is equipped with two objects: a source and a target. If a is the source for f and b is the target for f, then we may write $f : a \to b$ to denote the morphism f. The collection of morphisms $a \to b$ will be denoted* $\mathrm{Hom}_{\mathcal{C}}(a, b)$*.*

- *For any three objects a, b, c, there is a binary operation*

$$\mathrm{Hom}_{\mathcal{C}}(b, c) \times \mathrm{Hom}_{\mathcal{C}}(a, b) \to \mathrm{Hom}_{\mathcal{C}}(a, c)$$

  called composition. *If $f : a \to b$ and $g : b \to c$, we usually denote the composition of f and g by $g \circ f$.*

The composition satisfies two axioms:

- *If $f : a \to b$, $g : b \to c$, and $h : c \to d$ are morphisms, then $h \circ (g \circ f) = (h \circ g) \circ f$.*
- *For any object a, there is a morphism $1_a : a \to a$ called the* identity*, such that, for any morphism $f : a \to b$, we have $1_b \circ f = f = f \circ 1_a$.*

If $\mathcal{C}$ is a category, and $f : a \to b$ is a morphism in $\mathcal{C}$, then we call f an isomorphism *if there is a morphism $g : b \to a$ such that $f \circ g = \mathrm{id}_b$ and $g \circ f = \mathrm{id}_a$.*

Example 2.1.2.

- **Set** is the category whose objects are sets and whose morphisms are functions between sets.
- **Grp** is the category whose objects are groups and whose morphisms are group homomorphisms.
- **Ab** is the category whose objects are abelian groups and whose morphisms are group homomorphisms.
- **Ring** is the category whose objects are unital rings and whose morphisms are ring homomorphisms.
- **Top** is the category whose objects are topological spaces and whose morphisms are continuous functions.
- $\mathbf{Vect}_k$ is the category whose objects are finite dimensional vector spaces over a field k and whose morphisms are linear transformations.

Definition 2.1.3. *Let $\mathcal{C}$ be a category. The* opposite category *$\mathcal{C}^{op}$ is the category such that*

- $\mathrm{Ob}(\mathcal{C}^{op}) = \mathrm{Ob}(\mathcal{C})$*, and*
- *Given any two objects $a, b \in \mathrm{Ob}(\mathcal{C}^{op})$, we have $\mathrm{Hom}_{\mathcal{C}^{op}}(a, b) = \mathrm{Hom}_{\mathcal{C}}(b, a)$.*

If $f : a \to b$ and $g : b \to c$ are morphisms in $\mathcal{C}$ which define the morphisms $f' : b \to a$ and $g' : c \to b$ in $\mathcal{C}^{op}$, then we define the composition $f' \circ g'$ by

$$f' \circ g' := g \circ f.$$

Definition 2.1.4. *Let $\mathcal{C}$ and $\mathcal{D}$ be categories. A* covariant *(resp.* contravariant*)* functor *$F : \mathcal{C} \to \mathcal{D}$ consists of the following data:*

- *For each object a in $\mathcal{C}$, there is an object $F(a)$ in $\mathcal{D}$.*
- *For each morphism $f : a \to b$ in $\mathcal{C}$, there is a morphism $F(f) : F(a) \to F(b)$ (resp. $F(f) : F(b) \to F(a)$) in $\mathcal{D}$.*

Moreover, the following properties hold:

- *For every object a in $\mathcal{C}$, we have $F(1_a) = 1_{F(a)}$.*
- *For every morphism $f : a \to b$ and $g : b \to c$, we have $F(g \circ f) = F(g) \circ F(f)$ (resp. $F(g \circ f) = F(f) \circ F(g)$).*

Equivalently, a contravariant functor $F : \mathcal{C} \to \mathcal{D}$ is a covariant functor $F : \mathcal{C}^{op} \to \mathcal{D}$. For any category $\mathcal{E}$, there is an identity functor *$\mathrm{Id}_{\mathcal{E}} : \mathcal{E} \to \mathcal{E}$. This functor sends objects and morphisms to themselves. Let $F : \mathcal{C} \to \mathcal{D}$ be a functor. We call F an* isomorphism *if there is a functor $G : \mathcal{D} \to \mathcal{C}$ such that $F \circ G = \mathrm{Id}_{\mathcal{D}}$ and $G \circ F = \mathrm{Id}_{\mathcal{C}}$.*

Example 2.1.5. There is a covariant functor $F : \mathbf{Set} \to \mathbf{Grp}$ called the free functor. On objects, F sends a set X to the free group generated by X. On morphisms, F sends a set map $f : X \to Y$ to the unique group homomorphism $F(f) : F(X) \to F(Y)$ from the free group $F(X)$ to the free group $F(Y)$ induced by f.

Example 2.1.6. There is a covariant functor $F : \mathbf{Grp} \to \mathbf{Set}$ called the forgetful functor. On objects, F sends a group X to its underlying set. On morphisms, F sends a a group homomorphism $f : X \to Y$ to the corresponding set map $F(f)\colon F(X) \to F(Y)$.

Example 2.1.7. There is a covariant functor $F : \mathbf{Vect}_k \to \mathbf{Vect}_k$ called the double dual functor. On objects, F sends a vector space V to its dual vector space V^{**}. On morphisms, F sends a linear transformation $f : V \to W$ to the map $F(f) : V^{**} \to W^{**}$ given as follows. For each $g \in V^{**}$ and $h \in W^*$, we have $(F(f)(g))(h) = g(h \circ f)$.

Example 2.1.8. There is covariant functor $F : \mathbf{Grp} \to \mathbf{Ab}$ called the abelianization functor. On objects, F sends a group G to its abelianization $F(G) = G/[G, G]$, where $[G, G]$ is the commutator subgroup of G. On morphisms, F sends a group homomorphism $f : G \to H$ to the induced group homomorphism on the quotients $F(f) : F(G) \to F(H)$. Note that $F(f)$ is well-defined because $f([G, G]) \subseteq [H, H]$.

Definition 2.1.9. *Suppose F and G are covariant (resp. contravariant) functors from category $\mathcal{C}$ to category $\mathcal{D}$. A* natural transformation *$\eta : F \to G$ consists of the following data:*

- *For every object $a \in \mathcal{C}$, there is a morphism $\eta_a : F(a) \to G(a)$.*

- *For every morphism $f : a \to b$ in $\mathcal{C}$, we have $\eta_b \circ F(f) = G(f) \circ \eta_a$ (resp. $\eta_a \circ F(f) = G(f) \circ \eta_b$).*

We call a natural transformation $\eta : F \to G$ a natural isomorphism *if η_a is an isomorphism for every object a in $\mathcal{C}$.*

Example 2.1.10. Let $F : \mathbf{Grp} \to \mathbf{Ab}$ be the abelianization functor, and let $\mathrm{Id} : \mathbf{Grp} \to \mathbf{Grp}$ be the identity functor. There is a natural transformation $\pi : \mathrm{Id} \to F$. For any group G, we define the group homomorphism $\pi_G : G \to G/[G, G]$ by sending g to the image of g in the quotient group $G/[G, G]$.

Example 2.1.11. Let $F : \mathbf{Vect}_k \to \mathbf{Vect}_k$ be the double dual functor, and let $\mathrm{Id} : \mathbf{Vect}_k \to \mathbf{Vect}_k$ be the identity functor. There is a natural transformation $\mathrm{ev} : \mathrm{Id} \to F$. For any finite dimensional vector space V over k, we define the linear transformation $\mathrm{ev}_V : V \to V^{**}$ by sending a vector v to the linear functional $(f \mapsto f(v))$ for all $f \in V^*$.

Definition 2.1.12. *Let $\mathcal{C}$ be a category, and suppose $f : a \to c$ and $g : b \to c$ are two morphisms in $\mathcal{C}$. We call an object P of $\mathcal{C}$ the* pullback *of f and g if the following conditions hold:*

(1) there exists morphisms $p_1 : P \to a$ and $p_2 : P \to b$ such that the diagram

$$\begin{array}{ccc} P & \xrightarrow{p_2} & b \\ {\scriptstyle p_1}\downarrow & & \downarrow{\scriptstyle g} \\ a & \xrightarrow[f]{} & c \end{array}$$

commutes.

(2) Given morphisms $q_1 : Q \to a$ and $q_2 : Q \to b$ in $\mathcal{C}$, there exists a unique morphism $u : Q \to P$ such that

$$q_1 = p_1 \circ u \quad \text{and} \quad q_2 = p_2 \circ u.$$

2.2 Algebras and coalgebras

In this section, we discuss R-algebras and R-coalgebras, where R is a commutative unital ring. We closely follow Kassel [26, Part 1, §3] and Underwood [42, Ch. 4].

Definition 2.2.1. *An R-*algebra *is a triple $A = (A, \phi, \eta)$, where A is an R-module, and $\phi : A \otimes_R A \to A$ and $\eta : R \to A$ are R-linear maps such that the diagrams*

$$\begin{array}{ccc} A \otimes_R A \otimes_R A & \xrightarrow{\mathrm{id}_A \otimes_R \phi} & A \otimes_R A \\ {\scriptstyle \phi \otimes_R \mathrm{id}_A}\downarrow & & \downarrow{\scriptstyle \phi} \\ A \otimes_R A & \xrightarrow[\phi]{} & A \end{array} \qquad \textit{and} \qquad \begin{array}{ccccc} A \otimes_R R & \xrightarrow{\mathrm{id}_A \otimes_R \eta} & A \otimes_R A & \xleftarrow{\eta \otimes_R \mathrm{id}_A} & R \otimes_R A \\ & {\scriptstyle \simeq}\searrow & \downarrow{\scriptstyle \phi} & \swarrow{\scriptstyle \simeq} & \\ & & A & & \end{array}$$

commute. We call ϕ the product*, and we call η the* unit*.*

We say that A is a commutative *R-algebra if, in addition to the commutativity of the two diagrams above, the diagram*

$$\begin{array}{ccc} A \otimes_R A & \xrightarrow{\gamma} & A \otimes_R A \\ & {\scriptstyle\phi}\searrow \quad \swarrow{\scriptstyle\phi} & \\ & A & \end{array}$$

commutes, where $\gamma(a \otimes b) = b \otimes a$ is the switching map.

A morphism *$f : (A, \phi, \eta) \to (A', \phi', \eta')$ of R-algebras is an R-linear map $f : A \to A'$ such that*

$$\phi' \circ (f \otimes_R f) = f \circ \phi \quad \textit{and} \quad f \circ \eta = \eta'.$$

Example 2.2.2. If (A, ϕ_A, η_A) and (B, ϕ_B, η_B) are R-algebras, then the tensor product $A \otimes_R B$ has the structure of an R-algebra, with product

$$\phi((a_1 \otimes b_1) \otimes (a_2 \otimes b_2)) = \phi_A(a_1 \otimes a_2) \otimes \phi_B(b_1 \otimes b_2), \quad a_1, a_2 \in A, b_1, b_2 \in B,$$

and extended by R-linearity, and unit

$$\eta(r) = r \cdot (\eta_A(1) \otimes \eta_B(1)).$$

This algebra is commutative when A and B are commutative.

Definition 2.2.3. *An R-*coalgebra *is a triple $C = (C, \psi, \epsilon)$, where C is an R-module, and $\psi : C \to C \otimes_R C$ and $\epsilon : C \to R$ are R-linear maps such that the diagrams*

$$\begin{array}{ccc} C & \xrightarrow{\psi} & C \otimes_R C \\ {\scriptstyle\psi}\downarrow & & \downarrow{\scriptstyle \mathrm{id}_C \otimes_R \psi} \\ C \otimes_R C & \xrightarrow[\psi \otimes_R \mathrm{id}_C]{} & C \otimes_R C \otimes_R C \end{array} \quad \textit{and} \quad \begin{array}{ccccc} & & C & & \\ & {\scriptstyle\simeq}\swarrow & \downarrow{\scriptstyle\psi} & \searrow{\scriptstyle\simeq} & \\ C \otimes_R R & \xleftarrow[\mathrm{id}_C \otimes_R \epsilon]{} & C \otimes C & \xrightarrow[\epsilon \otimes_R \mathrm{id}_C]{} & R \otimes_R C \end{array}$$

commute. We call ψ the coproduct*, and we call ϵ the* counit*.*

We say that C is a cocommutative *R-coalgebra if, in addition to the commutativity of the two diagrams above, the diagram*

$$\begin{array}{ccc} & C & \\ & {\scriptstyle\psi}\swarrow \quad \searrow{\scriptstyle\psi} & \\ C \otimes_R C & \xrightarrow[\gamma]{} & C \otimes_R C \end{array}$$

commutes, where $\gamma(a \otimes b) = b \otimes a$ is the switching map.

A morphism *$f : (C, \psi, \epsilon) \to (C', \psi', \epsilon')$ of R-coalgebras is an R-linear map $f : C \to C'$ such that*

$$(f \otimes_R f) \circ \psi = \psi' \circ f \quad \textit{and} \quad \epsilon = \epsilon' \circ f.$$

Let (C, ψ, ϵ) be an R-coalgebra. Then, for any element $x \in C$, we can write

$$\psi(x) = \sum_i x_{(1)i} \otimes x_{(2)i}, \quad x_{(1)i}, x_{(2)i} \in C.$$

In order to avoid complicated index notation, we introduce *Sweedler notation*, which simply omits the index i:

$$\psi(x) = \sum_{(x)} x_{(1)} \otimes x_{(2)}.$$

Let M be an R-module. Then there is an R-module $M^* := \operatorname{Hom}(M, R)$ consisting of R-linear functionals on M. Suppose $g : M \to N$ is a map of R-modules. Then there is an R-module map $g^* : N^* \to M^*$, such that $g^*(f) = f \circ g$ for all $f \in N^*$. The map g^* is called the *transpose* of g.

Proposition 2.2.4. *If M and N are free finitely-generated R-modules, then the map*

$$q : M^* \otimes_R N^* \to (M \otimes_R N)^*,$$

defined by $q(f \otimes h)(a \otimes b) = f(a)h(b)$ for all $f \in M^*$, $h \in N^*$, $a \otimes b \in M \otimes_R N$, and extended by R-linearity, is an R-module isomorphism.

Proof: See [42, Prop. 4.1.6]. ∎

The proof of the following proposition can be found in [42, Prop. 4.1.7]. We reproduce the proof here.

Proposition 2.2.5. *Let (C, ψ, ϵ) be a cocommutative R-coalgebra. Then there is a commutative product with unit on C^*.*

Proof: The coproduct $\psi : C \to C \otimes_R C$ induces an R-linear map

$$\psi^* : (C \otimes_R C)^* \to C^*.$$

By Proposition 2.2.4, there is an isomorphism of R-modules

$$q : C^* \otimes_R C^* \to (C \otimes_R C)^*, \quad q(f \otimes h)(a \otimes b) = f(a)h(b).$$

Identifying $(C \otimes_R C)^*$ with $C^* \otimes_R C^*$ via q, we define the product ϕ on C^* as follows:

$$\phi(f \otimes h)(a) := \psi^*(f \otimes h)(a) = (f \otimes h)(\psi(a)) = \sum_{(a)} f(a_{(1)})h(a_{(2)}).$$

By the cocommutativity of (C, ψ, η), we see that $\phi(f \otimes h) = \phi(h \otimes f)$ for all $f, h \in C^*$. Set $\eta = \epsilon^*$. It is straightforward to verify that (C^*, ψ, η) is a commutative R-algebra. ∎

Let M be a free finitely generated R-module with basis $B = \{e_1, \ldots, e_n\}$. Then $M^* := \mathrm{Hom}(M, R)$ is a free finitely-generated R-module with basis $B^* = \{e_1^*, \ldots, e_n^*\}$, where $e_i^*(e_j) = \delta_{ij}$. Suppose that M has the structure of a cocommutative R-coalgebra, with coproduct ψ. Let $p^{\theta}_{\theta_1,\theta_2} \in R$ be the coefficients of the coproduct in the basis B, i.e., for every $e_k \in B$,

$$\psi(e_k) = \sum_{e_i, e_j \in B} p^{e_k}_{e_i,e_j} e_i \otimes e_j.$$

The proof of Proposition 2.2.4 implies that there is a commutative product ϕ on M^*, and the product satisfies

$$\phi({e_i}^* \otimes {e_j}^*) = \sum_{e_k \in B} p^{e_k}_{e_i,e_j} e_k^*.$$

2.3 Topological groups and rings

In this section, we will discuss topological group and topological rings. We closely follow Atiyah-Macdonald [1], Bourbaki [4] and [5], Husain [24], and Nagata [33].

Uniform spaces and topological groups are discussed at length in Bourbaki [4] and [5].

Definition 2.3.1. *Let X be a topological space. A nonempty collection Φ of subsets $U \subseteq X \times X$ is a* uniform structure *on X if it satisfies the following axioms:*

(1) If $U \in \Phi$, then $\Delta \subseteq U$, where $\Delta = \{(x, x) \mid x \in X\}$ is the diagonal of X.

(2) If $U \in \Phi$ and $U \subseteq V \subseteq X \times X$, then $V \in \Phi$.

(3) If $U, V \in \Phi$, then $U \cap V \in \Phi$.

(4) If $U \in \Phi$, then there is $V \in \Phi$ such that $V \circ V \subseteq U$, where

$$W \circ Z := \{(w, z) \in X \times X : \exists y \in X \text{ such that } (w, y) \in Z \text{ and } (y, z) \in W\}.$$

(5) If $U \in \Phi$, then $U^{-1} \in \Phi$, where $U^{-1} = \{(y, x) : (x, y) \in U\}$.

The elements U of Φ are called entourages*, and Φ is called a* uniformity *on X. The space X together with a uniformity Φ is called a* uniform space*. A* fundamental system *of entourages B of Φ is a set of entourages such that every element in Φ contains a set belonging to B.*

Example 2.3.2. Let (X, d) be a metric space. We can define a uniformity Φ on (X, d) as follows. Let $U \in \Phi$ if and only if U contains a subset of X of the form

$$U_r = \{(x, y) \in X \times X \mid d(x, y) \leq r\}, \quad r > 0.$$

Then Φ is a uniformity on X. Moreover, the subsets U_r, $r > 0$, form a fundamental system of entourages for this uniformity on X.

Suppose G is a topological space, endowed with the structure of a (not necessarily commutative) group such that the multiplication and inversion maps,

$$\mu : G \times G \to G, \quad (x, y) \mapsto xy \quad \text{and} \quad \iota : G \to G, \quad x \mapsto x^{-1},$$

are continuous. We call G a *topological group*. We denote the identity element of G by e. A *morphism* of topological groups is a continuous group homomorphism $\phi : G \to G'$.

Example 2.3.3. Let G be a topological group. We can define a uniformity Φ on G as follows. Let $U \in \Phi$ if and only if U contains a subset of the form

$$U_N = \{(x, y) \in G \times G \mid xy^{-1} \in N\},$$

where N is a neighbourhood of 0. Then Φ is a uniformity on G. Moreover, the subsets U_N form a fundamental system of entourages for this uniformity on G.

Definition 2.3.4. *Let X be a nonempty set, and let $\mathcal{F}$ be a nonempty subset of the powerset $\mathcal{P}(X)$. We call $\mathcal{F}$ a* filter *if it satisfies the following axioms:*

(1) $\emptyset \notin \mathcal{F}$;

(2) For all $F, G \in \mathcal{F}$, the intersection $F \cap G \in F$;

(3) Suppose $F \in \mathcal{F}$ and $H \in \mathcal{P}(X)$. If $F \subseteq H$, then $H \in \mathcal{F}$.

Definition 2.3.5. *A pair $(I, \leq)$ is called a* directed set *if I is a nonempty set and $\leq$ is a transitive, reflexive relation on I such that, for any $a, b \in I$, there is $c \in I$ such that $c \geq a$ and $c \geq b$.*

Definition 2.3.6. *Let X be a nonempty set, and let $(I, \leq)$ be a directed set. A* net *on X indexed by $(I, \leq)$ is a function $x : I \to X$. We usually denote x by $(x_i)_{i \in I}$, where $x_i := x(i)$.*

Let X be a nonempty set. The construction of derived nets and derived filters on X given below is taken out of [33, Def. II.12].

Suppose $(x_i)_{i \in I}$ is a net on X indexed by $(I, \leq)$. Set

$$\mathcal{F}_x := \{F \in X \mid (x_i)_{i \in I} \text{ is eventually in } F\}.$$

We call $\mathcal{F}_x$ the *filter derived* from the net $(x_i)_{i\in I}$.

Let I be any set in bijection with a filter $\mathcal{F}$. This bijection allows us to index the elements in $\mathcal{F}$ by the elements in I. Thus, $\mathcal{F} = \{F_i\}_{i\in I}$. We can view I as a directed set: we say that $i \leq j$ if and only if $F_i \supseteq F_j$. Any net

$$x_{\mathcal{F}} : I \to X, \quad x_{\mathcal{F}}(i) \in F_i$$

is called a *net derived* from the filter $\mathcal{F}$.

For the rest of this section, X will denote a uniform space with uniformity Φ. The discussion from here up to Theorem 2.3.12 is similar to the discussions in Husain [24, §29] and Nagata [33, Ch. II].

Let $\mathcal{F}$ be a filter in X. We say that $\mathcal{F}$ *converges* to $x \in X$ if, for every $U \in \Phi$, there is $F \in \mathcal{F}$ such that $(x_F, x) \in U$ for all $x_F \in F$. Now suppose $(x_i)_{i\in I}$ is a net on X indexed by a directed set $(I, \leq)$. We say that $(x_i)_{i\in I}$ *converges* to $x \in X$ if, for every $U \in \Phi$, there is $n \in I$ such that $(x_i, x) \in U$ for all $i \geq n$.

Theorem 2.3.7. *A filter $\mathcal{F}$ in X converges to $x \in X$ if and only if every net $x_{\mathcal{F}}$ derived from $\mathcal{F}$ converges to x.*

Proof: Suppose $\mathcal{F}$ converges to $x \in X$, and let I be an index set for $\mathcal{F}$. Let $U \in \Phi$. Then there exists $F_i \in \mathcal{F}$ such that $(x_{F_i}, x) \in U$ for all $x_{F_i} \in F_i$. If $j \geq i$, then $F_j \subseteq F_i$. Therefore, $(x_{F_j}, x) \in U$ for all $x_{F_j} \in F_j$ as well. Given a derived net $x_{\mathcal{F}}$, we have $x_{\mathcal{F}}(l) \in F_l$ for all $l \in I$. Therefore, $(x_{\mathcal{F}}(j), x) \in U$ for all $j \geq i$. So $x_{\mathcal{F}}$ converges to x.

Now assume that $\mathcal{F}$ does not converge to x. Then there is some entourage $U \in \Phi$ such that, for all $F_i \in \mathcal{F}$, there is $x_{F_i} \in F_i$ with $(x_{F_i}, x) \notin U$. Thus, the derived net $x_{\mathcal{F}}$ defined by $x_{\mathcal{F}}(i) = x_{F_i}$ does not converge to x. ∎

Theorem 2.3.8. *A net $x : I \to X$ converges to $x \in X$ if and only if its derived filter $\mathcal{F}$ does.*

Proof: Suppose that x converges to $y \in X$. Then, for every $U \in \Phi$, there is $n \in I$ with $(x_i, y) \in U$ for all $i \geq n$. Let $F = \{x_i \mid i \geq n\} \in \mathcal{F}_x$. Then, for all $x_F \in F$, we have $(x_F, y) \in U$. So $\mathcal{F}_x$ converges to y.

Now suppose that $\mathcal{F}_x$ converges to $y \in Y$. Then, for every $U \in \Phi$, there is $F \in \mathcal{F}_x$ such that $(x_F, y) \in U$ for all $x_F \in F$. By definition of $\mathcal{F}_x$, there is $n \in I$ such that, for all $i \geq n$, we have $x_i \in F$. Thus $(x_i, y) \in U$ for all $i \geq n$, so x converges to y. ∎

A net $(x_i)_{i\in I}$ on X is a *Cauchy net* if, for any entourage $U \in \Phi$, there exists $j \in I$ such that for all $i, k \geq j$, we have $(x_i, x_k) \in U$. A filter $\mathcal{F}$ on X is a *Cauchy filter* if, for any entourage $U \in \Phi$, there is $F \in \mathcal{F}$ such that $(x_F, y_F) \in U$ for all x_F, $y_F \in F$.

Proposition 2.3.9. *Suppose a filter $\mathcal{F}$ in X is Cauchy. Then every derived net $x_{\mathcal{F}}$ is Cauchy.*

Proof: Since $\mathcal{F}$ is Cauchy, for any $U \in \Phi$, there is $F_i \in \mathcal{F}$ such that $(x_{F_i}, y_{F_i}) \in U$ for all $x_{F_i}, y_{F_i} \in F_i$. Recall that $j \geq i$ whenever $F_j \subseteq F_i$. Thus, given a derived net $x_{\mathcal{F}}$, for all $j, k \geq i$, we have $(x_{\mathcal{F}}(j), x_{\mathcal{F}}(k)) \in U$. So every derived net $x_{\mathcal{F}}$ is a Cauchy net. ∎

Proposition 2.3.10. *Suppose a net $(x_i)_{i \in I}$ is Cauchy. Then the derived filter $\mathcal{F}_x$ is Cauchy.*

Proof: Since $(x_i)_{i \in I}$ is a Cauchy net, for any $U \in \Phi$, there is $n \in I$ such that $(x_i, x_j) \in U$ for all $i, j \geq n$. Set $F = \{x_i \mid i \geq n\}$. Then $F \in \mathcal{F}_x$, and, for every $x_i, x_j \in F$, we have $(x_i, x_j) \in U$. So $\mathcal{F}_x$ is a Cauchy filter. ∎

Theorem 2.3.7, Theorem 2.3.8, Proposition 2.3.9, and Proposition 2.3.10 imply that all Cauchy filters on X converge if and only if all Cauchy nets on X converge.

Definition 2.3.11. *A uniform space X is complete if every Cauchy net (or Cauchy filter) in X converges.*

Theorem 2.3.12. *Suppose X is a metrizable topological space. Let Φ be the standard uniformity on the metric space X. The space X is complete as a uniform space if and only if it is complete as a metric space.*

Proof: See the proof of [24, §29, Theorem 6]. ∎

We now discuss topological groups and rings. We closely follow Bourbaki [4], [5].

Lemma 2.3.13. *Let G be a topological group, and let N be a normal subgroup in G. The topological closure $\overline{N}$ of N is a normal subgroup in G.*

Proof: Let $x, y \in \overline{N}$. The element $xy^{-1} \in \overline{N}$ by [4, Ch. III, §2.1, Prop. 1]. ∎

Theorem 2.3.14. *Suppose G is a commutative metrizable complete Hausdorff topological group, and N is a closed normal subgroup in G. Then the quotient G/N is a complete Hausdorff topological group.*

Proof: Since G is a commutative metrizable complete Hausdorff topological group, and N is a closed normal subgroup of G, the quotient group G/N is complete by [5, Ch. IX, §3, no. I, Prop. 4] and Hausdorff by [4, Ch. III, §2.6, Prop. 18a]. ∎

We now state and prove some useful facts about topological rings. Suppose R is a topological space, endowed with the structure of a ring such that the addition and multiplication maps,

$$a : R \times R \to R, \quad (x, y) \mapsto x + y \quad \text{and} \quad \mu : R \times R \to R, \quad (x, y) \mapsto xy,$$

are continuous. We call S a *topological ring*. It follows from the definition that R is a topological group with respect to addition and subtraction. We say that the topological ring R is *complete* if and only if it is complete as a topological group. A *morphism* $\phi : R \to R'$ of topological rings is a continuous ring homomorphism.

Lemma 2.3.15. *Let R be a topological ring, and let I be an ideal in R. The topological closure $\overline{I}$ of I is an ideal in R.*

Proof: Let $x, y \in \overline{I}$ and $s \in S$. The fact that $x - y \in \overline{I}$ follows from Lemma 2.3.13. To see that $sx \in \overline{I}$, let U be a neighbourhood of sx. Since $x \in \overline{I}$, every neighbourhood Y of x intersects I nontrivially. The inverse image of U under the multiplication map $(a, b) \mapsto ab$ is open in $S \times S$ by continuity of multiplication, and contains (s, x). Thus, there is a neighbourhood Y of x such that $(s \cdot Y) \cap I$ is nonempty and $(s \cdot Y) \subseteq U$. So $U \cap I$ is nonempty, as required. ∎

Theorem 2.3.16. *Suppose R is a metrizable complete Hausdorff topological ring, and I is a closed ideal in R. Then the quotient R/I is a complete Hausdorff topological ring.*

Proof: This follows from Theorem 2.3.14. ∎

For the rest of this section, we follow Atiyah-Macdonald [1]. Let S be a ring, and let I be an ideal in S. The I-*adic* topology on S is the topology generated by elements of the form $x + I^n$, where $x \in S$ and $n \geq 1$.

Lemma 2.3.17. *The ring S is a topological ring in the I-adic topology.*

Proof: Let U be a neighbourhood of $x \in S$ in the I-adic topology. Then U contains $x + I^n$ for some $n \geq 1$.

The pair $(x + I^n, I^n)$ is contained in the inverse image of $x + I^n$ under addition. Since $(x + I^n, I^n)$ is open in the product topology on $S \times S$, we see that addition is continuous in the I-adic topology.

The pair $(x + I^n, 1 + I^n)$ is contained in the inverse image $x + I^n$ under multiplication. Since $(x + I^n, 1 + I^n)$ is open in the product topology on $S \times S$, we see that multiplication is continuous in the I-adic topology. ∎

The following two facts are proven in [1, Ch. 10]:

- The ring S is Hausdorff in the I-adic topology if and only if $\cap_{i\geq 0} I^i = (0)$.
- The ring S is complete in the I-adic topology if and only if the canonical ring homomorphism $S \to \varprojlim_i (S/I^i)$ is an isomorphism.

Suppose S is a Hausdorff topological ring in the I-adic topology. Then S is metrizable, and the metric inducing this topology can be defined as follows. Given $s \in S \setminus \{0\}$, set $\operatorname{ord}_I(s) = t$, where $t \geq 0$ is the largest integer such that $s \in I^t$ (here, we take $I^0 := S$), and $\operatorname{ord}_I(0) = \infty$. For any $s_1, s_2 \in S$, we define the *distance* $d_I(s_1, s_2) := 2^{-\operatorname{ord}_I(s_1 - s_2)}$. The topology induced by this metric coincides with the I-adic topology on S. Hence, S is metrizable. If, in addition to being Hausdorff, S is complete, then Theorem 2.3.16 implies that S/J is Hausdorff and complete, where J is any closed ideal in S.

Example 2.3.18. Let R be a commutative unital ring. Let I be the ideal $(x_1, \ldots, x_n)$ in the polynomial ring $R[x_1, \ldots, x_n]$. The I-adic completion of $R[x_1, \ldots, x_n]$ is isomorphic to the ring of formal power series $R[\![x_1, \ldots, x_n]\!]$. Moreover, $R[\![x_1, \ldots, x_n]\!]$ is a complete Hausdorff ring in the $\overline{I}$-adic topology, where $\overline{I}$ is the topological closure of the image of I in $R[\![x_1, \ldots, x_n]\!]$.

Chapter 3

Oriented cohomology theories

In this chapter, we fix once and for all an algebraically closed field k of characteristic 0. The goal of this chapter is to motivate and define (algebraic) oriented cohomology theories in the sense of Levine-Morel [31]. In Section 3.1, we discuss one-dimensional commutative formal group laws. In Section 3.2, we discuss algebraic varieties over k. The definition of an algebraic variety that we use in this chapter is given in Definition 3.2.1. In Section 3.3, we introduce vector bundles over algebraic varieties. In Section 3.4, we introduce the Chow theory, which is the most basic example of an oriented cohomology theory. In Section 3.5, we introduce the concept of an oriented cohomology theory. The main references that we use in this chapter are Atiyah-Macdonald [1], Fulton [15], Hartshorne [18], Hazewinkel [19], Humphreys [22], Levine-Morel [31], and Shafarevich [35] and [36].

3.1 Formal group laws

In this section, we fix a commutative unital ring R. The goal of this section is to introduce the concept of a one-dimensional commutative formal group law. These formal group laws are closely related to oriented cohomology theories. The relationship between these formal group laws and oriented cohomology theories will be emphasized in the upcoming sections. We closely follow Hazewinkel [19] in this section.

Definition 3.1.1. *A* one-dimensional commutative formal group law *over R is a power series $F = F(u, v) \in R[\![u, v]\!]$ satisfying the following axioms:*

(1) $F(0, u) = F(u, 0) = u \in R[\![u]\!]$,

(2) $F(u, v) = F(v, u)$, *and*

(3) $F(u, F(v, w)) = F(F(u, v), w) \in R[\![u, v, w]\!]$.

From now one, we will take "formal group law" to mean "one-dimensional commutative formal group law." Let $F(u,v) \in R[\![u,v]\!]$ be a formal group law. Then there are coefficients $a_{i,j} \in R$ such that

$$F(u,v) = \sum_{i,j\geq 0} a_{i,j} u^i v^j.$$

The coefficients $a_{i,j}$ have some restrictions placed on them by the the axioms of the formal group law. For example, axiom (1) implies that $a_{0,0} = 0$ and that $a_{1,0} = a_{0,1} = 1$. Axiom (2) implies that $a_{i,j} = a_{j,i}$ for all $i, j \geq 0$. The relations imposed by axiom (3) are much more complicated.

The *formal inverse* of u is the unique power series in $R[\![u]\!]$, denoted by $-_F u$, such that $F(u, -_F u) = 0$. We can write

$$-_F u = -\sum_{i\geq 1} c_i u^i$$

for some coefficients $c_i \in R$. The coefficients c_i can be computed in terms of the coefficients $a_{i,j}$ through the equation $F(u, -_F u) = 0$. The output of the Python program in Appendix A gives the first six coefficients c_i:

$$\begin{aligned}
c_1 &= 1;\\
c_2 &= -a_{11};\\
c_3 &= -a_{11}c_2;\\
c_4 &= -a_{11}c_3 + a_{12}c_2 - 2a_{13} + a_{22};\\
c_5 &= -a_{11}c_4 + a_{12}c_3 + a_{12}c_2^2 - 4a_{13}c_2 + 2a_{22}c_2;\\
c_6 &= -a_{11}c_5 + a_{12}c_4 + 2a_{12}c_2c_3 - 4a_{13}c_3 - 3a_{13}c_2^2 + 3a_{14}c_2 - 2a_{15} + 2a_{22}c_3\\
&\quad + a_{22}c_2^2 - a_{23}c_2 + 2a_{24} - a_{33}.
\end{aligned}$$

Example 3.1.2. The *additive* formal group law over R is the series

$$F_A(u,v) = u + v.$$

In this case $-_A u = -u$.

Example 3.1.3. Let $\beta \in R \setminus \{0\}$. The *multiplicative* formal group law over R with respect to β is the series

$$F_M(u,v) = u + v - \beta uv.$$

In this case, $-_M u = -\sum_{i\geq 0} \beta^i u^i$. If $\beta \in R^\times$, we say that F_M is *multiplicative periodic*.

Example 3.1.4. Let $\beta \in R \setminus \{0\}$. The *Lorentz* formal group law over R with respect to β is the series

$$F_L(u,v) = \frac{u+v}{1+\beta uv} = (u+v)\sum_{i\geq 0}(\beta uv)^i.$$

In this case, $-_L u = -u$.

Example 3.1.5. Let $\mathbb{L}$ be the *Lazard ring*, i.e., the commutative unital ring with generators $a_{i,j}$, $i, j \geq 1$, subject only to the relations imposed by the axioms of the formal group law. The *universal* formal group law is the series

$$F_U(u,v) = u + v + \sum_{i,j\geq 1} a_{i,j} u^i v^j \in \mathbb{L}[\![u,v]\!].$$

The formal group law F_U is *universal* in the following sense:

Theorem 3.1.6. *For any formal group law F over R, there is a unique ring homomorphism*

$$f : \mathbb{L} \to R$$

such that $F(u,v) = u + v + \sum_{i,j\geq 1} f(a_{i,j}) u^i v^j$.

Proof: Write $F(u,v) = u + v + \sum_{i,j\geq 1} b_{i,j} u^i v^j$ for some $b_{i,j} \in R$. Define f by the rule $f(a_{i,j}) = b_{i,j}$ and extend by linearity. This map is well-defined (i.e., it preserves the relations in the Lazard ring) by definition of the Lazard ring. Finally, we show uniqueness. Since

$$u + v + \sum_{i,j\geq 1} f(a_{i,j}) u^i v^j = u + v + \sum_{i,j\geq 1} b_{i,j} u^i v^j \in R[\![u,v]\!],$$

we must have $f(a_{i,j}) = b_{i,j}$ for all $i, j \geq 1$. ∎

3.2 Algebraic varieties

In this section, we discuss algebraic varieties over k. We closely follow Fulton [15], Hartshorne [22], Humphreys [22], and Shafarevich [35] and [36].

Affine n-space is the n-tuple k^n. An *affine algebraic set* in $\mathbb{A}^n$ is the set of common zeros in $\mathbb{A}^n$ of a finite number of polynomials in n variables. The *Zariski topology* on $\mathbb{A}^n$ is topology on $\mathbb{A}^n$ whose closed sets are precisely the affine algebraic sets. An *affine variety* is an irreducible closed subset of $\mathbb{A}^n$ whose topology is induced by the Zariski topology on $\mathbb{A}^n$. Recall that a topological space X is *Noetherian* if, for any sequence $Y_1 \supseteq Y_2 \supseteq \cdots$ of closed subsets Y_i of X, there is $m > 0$ such that $Y_m = Y_{m+1} = \cdots$. Any affine variety is Noetherian. A *quasi-affine variety* is an open subset of an affine variety, with the induced topology. If X is an affine variety, then the topology on X is generated by the open subsets

$$D_f := \{x \in X \mid f(x) \neq 0\},$$

where $f \in k[x_1, \dots, x_n]$. The set

$$I(X) := \{f \in k[x_1, \dots, x_n] \mid f(x) = 0 \text{ for all } x \in X\}$$

is an ideal, and $I(X)$ called the *ideal of vanishing* on X. The *coordinate ring* of X is the quotient ring

$$k[X] := k[x_1, \dots, x_n]/I(X).$$

Let Y be a quasi-affine variety. A function $f : Y \to k$ is *regular at a point* $P \in Y$ if there is an open neighbourhood U of P in Y such that $f = \frac{g}{h}$ on U, where $g, h \in k[x_1, \dots, x_n]$ and h does not vanish along U. The function f is *regular on* Y if it is regular at every point on Y. If f is regular on Y, then f is continuous.

Projective n-space $\mathbb{P}^n$ is the set of equivalence classes of $k^{n+1} \setminus \{0\}$ with respect to the equivalence relation

$$(x_0, \dots, x_n) \sim (y_0, \dots, y_n)$$

if and only if there exists $\lambda \in k$ such that, for all $i = 0, \dots, n$, we have $y_i = \lambda x_i$. A *projective algebraic set* in $\mathbb{P}^n$ is the set of common zeros in $\mathbb{P}^n$ of a finite number of *homogeneous* polynomials in $n+1$ variables. The *Zariski topology* on $\mathbb{P}^n$ is the topology on $\mathbb{P}^n$ whose closed sets are precisely the projective algebraic sets. A *projective variety* is an irreducible closed subset of $\mathbb{P}^n$ whose topology is induced by the Zariski topology on $\mathbb{P}^n$. Observe that a projective variety is Noetherian. A *quasi-projective variety* is an open subset of a projective variety. If X is a projective variety, then the set

$$I(X) := \{f \in k[x_0, \dots, x_n] \mid f(x) = 0 \text{ for all } x \in X \text{ and } f \text{ is homogeneous}\}$$

is an ideal in $k[x_0, \dots, x_n]$, and $I(X)$ is called the *ideal of vanishing* on X. The *homogeneous coordinate ring* of X is the quotient ring

$$k[X] := k[x_0, \dots, x_n]/I(X).$$

Let H_i be the zero set of x_i in $\mathbb{P}^n$, and let U_i be the open set $\mathbb{P}^n \setminus H_i$. Then $\mathbb{P}^n = \cup_{i=0}^n U_i$, and there is a homeomorphism

$$\phi_i : U_i \to \mathbb{A}^n, \quad \phi_i(a_0, \dots, a_n) = (\tfrac{a_0}{a_i}, \dots, \tfrac{a_n}{a_i}),$$

where the i-th coordinate is omitted. The inverse of ϕ_i is the map

$$\beta_i : \mathbb{A}^n \to U_i, \quad \beta_i(a_1, \dots, a_n) = (a_1, \dots, 1, \dots, a_n),$$

where the 1 appears as the i-th coordinate. In particular, we can view any affine or quasi-affine variety as an open subset of a projective variety, via a homeomorphism ϕ_i. Let Y be a quasi-projective variety. A function $f : Y \to k$ is *regular at a point*

$P \in Y$ if there is an open neighbourhood U of P in Y such that $f = \frac{g}{h}$ on U, where $g, h \in k[x_0, \ldots, x_n]$ are homogeneous polynomials of the same degree, and such that h does not vanish along U. The function f is *regular on* Y if it is regular at every point in Y. If f is regular on Y, then f is continuous.

Definition 3.2.1. *An* algebraic variety over k *is any quasi-projective variety over* k*.*

Let X and Y be algebraic varieties over k. A *morphism* of algebraic varieties $\phi : X \to Y$ is a continuous map such that, for any open set $U \subseteq Y$, and for any regular function $f : U \to k$, the function $f \circ \phi : \phi^{-1}(U) \to k$ is regular. Let $\phi : X \to Y$ be a morphism of affine varieties. There is a ring homomorphism

$$\phi^\star : k[Y] \to k[X], \quad \phi^\star(f) = f \circ \phi.$$

Definition 3.2.2. *Let* X *be an algebraic variety. The* dimension $\dim(X)$ *of* X *is the supremum over the length* n *of a chain of distinct subvarieties* $Z_0 \subseteq Z_1 \subseteq \cdots \subseteq Z_n$ *in* X*. If* Y *is an algebraic subvariety of an algebraic variety* X*, then the* codimension $\mathrm{codim}_X(Y)$ *of* Y *in* X *is the supremum over the length* n *of a chain of distinct subvarieties* $Y = Z_0 \subseteq Z_1 \subseteq \cdots \subseteq Z_n$ *in* X *beginning with* Y*. If* $X = \mathbb{P}^n$ *or* $X = \mathbb{A}^n$*, then we use the notation* $\mathrm{codim}(Y) := \mathrm{codim}_X(Y)$*.*

An algebraic variety X over k is *complete* if, for all algebraic varieties Y over k, the projection map $X \times Y \to Y$ is closed. The following facts concerning complete varieties are proven in [22, Ch. I]:

Proposition 3.2.3. *(1) Projective varieties are complete.*

(2) If $\phi : X \to Y$ *is a morphism of algebraic varieties, and* X *is complete, then the image of* ϕ *in* Y *is complete.*

We remark that the projection map from $\mathbb{P}^n \times \mathbb{P}^m$ onto either factor is closed by Proposition 3.2.3. It is straightforward to verify that this projection map is continuous.

Let $\sigma : \mathbb{P}^n \times \mathbb{P}^m \to \mathbb{P}^{(n+1)(m+1)-1}$ be the map sending

$$([x_0 : \cdots : x_n], [y_0 : \cdots : y_m]) \mapsto [x_0y_0 : x_0y_1 : \cdots : x_iy_j : \cdots x_ny_m].$$

The map σ is called the *Segre embedding*. The Segre embedding is well-defined and injective, and its image is closed in $\mathbb{P}^{(n+1)(m+1)-1}$ (see [18, Ch. I, Exercise 2.14]). If X is a subvariety of $\mathbb{P}^n$ and Y is a subvariety of $\mathbb{P}^m$, then $\sigma(X \times Y)$ is a subvariety of $\sigma(\mathbb{P}^n \times \mathbb{P}^m)$ (see [22, Ch. I, §1.7]).

Lemma 3.2.4. *Let* $X \subset \mathbb{P}^n$ *and* $Y \subseteq \mathbb{P}^m$ *be algebraic varieties over* k*. The Cartesian product* $X \times Y$ *is an algebraic variety over* k*.*

Proof: Since quasi-projective varieties are locally closed, we write $X = U_1 \cap Z_1$ and $Y = U_2 \cap Z_2$, where $U_1 \in \mathbb{P}^n$ and $U_2 \in \mathbb{P}^m$ are open and $Z_1 \in \mathbb{P}^n$ and $Z_2 \in \mathbb{P}^m$ are closed. The projection maps $\pi_1 : \mathbb{P}^n \times \mathbb{P}^m \to \mathbb{P}^n$ and $\pi_2 : \mathbb{P}^n \times \mathbb{P}^m \to \mathbb{P}^m$ are closed and continuous. We have $\pi_1^{-1}(X) = X \times \mathbb{P}^m$ and $\pi_2^{-1}(Y) = \mathbb{P}^n \times Y$. Thus,

$$\begin{aligned} X \times Y = \pi_1^{-1}(X) \cap \pi_2^{-1}(Y) &= (\pi_1^{-1}(U_1) \cap \pi_1^{-1}(Z_1)) \cap (\pi_2^{-1}(U_2) \cap \pi_2^{-1}(Z_2)) \\ &= (\pi_1^{-1}(U_1) \cap \pi_2^{-1}(U_2)) \cap (\pi_1^{-1}(Z_1) \cap \pi_2^{-1}(Z_2)). \end{aligned}$$

In particular,

$$\sigma(X \times Y) = \sigma((\pi_1^{-1}(U_1) \cap \pi_2^{-1}(U_2))) \cap \sigma((\pi_1^{-1}(Z_1) \cap \pi_2^{-1}(Z_2)))$$

is locally closed.

Next, we will show that $\sigma(X \times Y)$ is irreducible using the technique used in the the proof of [22, Ch. 1, Proposition 1.4]. Assume X and Y are irreducible, and assume that $X \times Y = Z_1 \cup Z_2$, where Z_1 and Z_2 are closed subsets of $X \times Y$. Let $x = [x_0 : \cdots : x_n] \in X$. Then $\sigma(\{[x_0 : \cdots : x_n]\} \times Y)$ is the closed subvariety of $\mathbb{P}^{(n+1)(m+1)-1}$ consisting of the points

$$[x_0y_0 : x_0y_1 : \cdots : x_iy_j : \cdots : x_ny_m], \quad \text{where} \quad [y_0 : \cdots : y_m] \in \mathbb{P}^m.$$

There must exist some i such that x_i is nonzero. Without loss of generality, assume x_0 is nonzero. It is easy to see that the map $\sigma(\{[x_0 : \cdots : x_n]\} \times Y) \to Y$ given by

$$[z_1 : \cdots : z_{(n+1)(m+1)}] \mapsto [z_1 : \cdots : z_{m+1}]$$

is an isomorphism of varieties over k, with inverse

$$[y_0 : \cdots : y_m] \mapsto [x_0y_0 : x_0y_1 : \cdots : x_iy_j : \cdots : x_ny_m].$$

Thus $\{x\} \times Y$ is irreducible.

Write $X = X_1 \cup X_2$ with $X_i = \{x \in X \mid \{x\} \times Y \subseteq Z_i\}$. Given $y \in Y$, the set $X \times \{y\}$ is closed, so the intersection $(X \times \{y\}) \cap Z_i$ is closed as well. Since the projection π onto the first factor is a closed map, the set $X_y^{(i)}$ of first coordinates is closed in X. In particular, the intersection $X_i = \cap_{y \in Y} X_y^i$ is closed. By irreducibility, we must have $X = X_1$ or $X = X_2$. By definition of the X_i, we therefore have that $X \times Y = Z_1$ or $X \times Y = Z_2$. ∎

Remark 3.2.5. The product of algebraic varieties $X \times Y$ is the product of X and Y in the category of algebraic varieties over k.

Let X be an algebraic variety over k, and let $\mathcal{O}(X)$ be the *ring of regular functions* on X. If X is an affine variety, then there is an isomorphism $\mathcal{O}(X) \simeq k[X]$.

Definition 3.2.6. *Let P be a point in X. The* local ring $\mathcal{O}_{X,P}$ *of X at P is the ring generated by equivalence classes of pairs $\langle U, f\rangle$, where U is an open subset of X containing P, f is a regular function on U, and $\langle U, f\rangle$ is equivalent to another pair $\langle V, g\rangle$ if and only if $f = g$ on the intersection $U \cap V$. Its maximal ideal consists pairs $\langle U, f\rangle$, such that f vanishes at P.*

Definition 3.2.7. *Let Y be a subvariety of X. The* local ring $\mathcal{O}_{X,Y}$ *of X along the subvariety Y is the ring generated by equivalence classes of pairs $\langle U, f\rangle$, where U is a nonempty open subset of Y, f is a regular function on U, and $\langle U, f\rangle$ is equivalent to another pair $\langle V, g\rangle$ if and only if $f = g$ on the intersection $U \cap V$. Its maximal ideal consists pairs $\langle U, f\rangle$, such that f vanishes on U.*

Definition 3.2.8. *The* function field $k(X)$ *of X is the field generated by equivalence classes of pairs $\langle U, f\rangle$, where U is a nonempty open subset of X, f is a regular function on U, and $\langle U, f\rangle$ is equivalent to another pair $\langle V, g\rangle$ if and only if $f = g$ on the intersection $U \cap V$. The elements of $k(X)$ are called* rational functions *on X.*

Definition 3.2.9. *A* rational map $f : X \to Y$ *between algebraic varieties X and Y is an equivalence class of pairs (f_U, U), where $U \subseteq X$ is a nonempty open set and $f_U : U \to Y$ is a morphism of algebraic varieties, where two pairs (f_U, U) and $(f'_{U'}, U')$ are equivalent whenever $f_U = f'_{U'}$ on the intersection $U \cap U'$. The morphism f is* birational *if there is a rational map $g : Y \to X$ which is inverse to f.*

The following three technical definitions will be used frequently in this chapter.

- A ring homomorphism $A \to B$ is *flat* if every exact sequence of A-modules remains exact after tensoring over A with B (see [15, Appendix A.4]). A morphism of algebraic varieties $f : X \to Y$ is *flat* if, for any affine open subsets $U \subseteq Y$ and $U' \subseteq X$ with $f(U') \subseteq U$, the induced map $f^* : k[U] \to k[U']$ makes $k[U']$ into a flat $k[U]$-module (see [15, Appendix B.2.5]). A morphism of algebraic varieties $f : X \to Y$ has *relative dimension n* if for all subvarieties V of Y, and all irreducible components V' of $f^{-1}(V)$, we have $\dim(V') = \dim(V) + n$ (see [15, Appendix B.2.5]).

- A morphism of algebraic varieties $f : X \to Y$ is *proper* if it is a composition $f = p \circ i$, where $i : X \to \mathbb{P}^n \times Y$ is a closed embedding and $p : \mathbb{P}^n \times Y \to Y$ is the projection onto the second factor. See [35, pp. 59].

- Roughly speaking, the *tangent space* of an algebraic variety X at a point x is set of lines tangent in X to the point x. We denote the tangent space at x by Θ_x. Let $\mathfrak{m}_x$ be maximal ideal of the local ring $\mathcal{O}_{X,x}$. The ideal $\mathfrak{m}_x$ is generated by equivalence classes of regular functions that vanish at x. As vector spaces, it is possible to identify Θ_x with the dual vector space $(\mathfrak{m}_x/\mathfrak{m}_x^2)^*$ over the field $K = \mathcal{O}_{X,x}/\mathfrak{m}_x$ (see [35, Ch. II, §1.3, Theorem 2]). If the dimension of Θ_x is constant over all $x \in X$, we say that X is *nonsingular* or *smooth*.

Remark 3.2.10. The product of smooth algebraic varieties over k is smooth (see [22, Ch. I, Proposition 5.1]).

3.3 Vector bundles

In this section, we discuss vector bundles over algebraic varieties. We closely follow Fulton [15], Shafarevich [35] and [36], and Kahanpää-Kekäläinen-Smith-Traves [38].

Definition 3.3.1. *Let E and X be algebraic varieties over k. We call a morphism of algebraic varieties $\pi : E \to X$ a* vector bundle of rank n *if the following conditions hold:*

(1) For all $x \in X$, the fiber $\pi^{-1}(x)$ is a finite-dimensional vector space over k.

(2) There is an open covering $\{U_i\}$ of X and isomorphisms of algebraic varieties $\phi_i : \pi^{-1}(U_i) \to U_i \times \mathbb{A}^n$, such that $p = \pi \circ \phi_i^{-1}$ on $U_i \times \mathbb{A}^n$, where $p : U_i \times \mathbb{A}^n \to U_i$ is the canonical projection onto the first factor.

A morphism *of vector bundles over X, $\pi : E \to X$ to $\pi' : E' \to X$, is a morphism of algebraic varieties $f : E \to E'$ such that $\pi = \pi' \circ f$, and $f|_{\pi^{-1}(x)} : \pi^{-1}(x) \to \pi'^{-1}(x)$ is linear for all $x \in X$.*

If a morphism of algebraic varieties $\pi : E \to X$ satisfies axiom (2), we say that $\pi : E \to X$ is *locally trivial*. In this case, the ϕ_i are called *local trivializations*. If $\pi : E \to X$ is a vector bundle, we call E the *total space* of the vector bundle. A vector bundle of rank 1 is called a *line bundle*. There is an alternative way to define vector bundles:

Definition 3.3.2. *Let $\pi : E \to X$ be a morphism of algebraic varieties over k. We call $\pi : E \to X$ a* vector bundle of rank n *if there is an open covering $\{U_i\}$ of X, and isomorphisms of algebraic varieties $\phi_i : \pi^{-1}(U_i) \to U_i \times \mathbb{A}^n$ such that, on $U_i \cap U_j$, we have*

$$\phi_i \circ \phi_j^{-1}(x, v) = (x, g_{i,j}(x)(v)),$$

where

$$g_{i,j} : U_i \cap U_j \to \mathrm{GL}(r, k)$$

is a morphism of algebraic varieties over k satisfying:

$$g_{i,j} = g_{i,j} g_{j,k}, \quad g_{i,j} = g_{j,i}^{-1}, \quad g_{i,i} = 1.$$

We call the $g_{i,j}$ transition functions. *Here, $\mathrm{GL}(r, k)$ is the group of invertible $r \times r$ matrices with entries in k, viewed as an algebraic variety over k (see Chapter 4).*

Lemma 3.3.3. *Let X be an algebraic variety over k, with open covering $\{U_i\}$. If $\pi : E \to X$ and $\pi' : E' \to X$ are two vector bundles over X with the same transition functions $g_{i,j}$, then there is an isomorphism of vector bundles $f : E \to E'$.*

Proof: Let

$$\phi_i : \pi^{-1}(U_i) \to U_i \times \mathbb{A}^r \quad \text{and} \quad \psi_i : \pi'^{-1}(U_i) \to U_i \times \mathbb{A}^r$$

be the respective local trivializations. Since the transition functions of the two vector bundles are the same, on an intersection $U_i \cap U_j$, we have

$$\phi_i \circ \phi_j^{-1} = \psi_i \circ \psi_j^{-1} \implies \psi_i^{-1} \circ \phi_i = \psi_j^{-1} \circ \phi_j.$$

Define a map $f : E \to E'$ as follows. Let $x \in E$. Since $\{U_i\}$ covers X and π is surjective, there exists U_i such that $x \in \pi^{-1}(U_i)$. Define $f(x) = \psi_i^{-1} \circ \phi_i(x)$. This map is independent of the index i: if $x \in \pi^{-1}(U_i \cap U_j)$, then

$$\psi_i^{-1} \circ \phi_i(x) = \psi_j^{-1} \circ \phi_j(x).$$

Moreover, the map f is a morphism of algebraic varieties because the ϕ_i and the ψ_i are morphisms of algebraic varieties. On $\pi^{-1}(U_i)$, we have

$$\pi' \circ f = \pi' \circ \psi_i^{-1} \circ \phi_i = p \circ \phi_i = \pi,$$

where $p : U_i \times \mathbb{A}^r \to U_i$ is projection onto the first factor. Finally, $f|_{\pi^{-1}(x)} : \pi^{-1}(x) \to \pi'^{-1}(x)$ is linear for all $x \in X$, since the $\phi_i|_{\pi^{-1}(x)}$ and $\psi_i|_{\pi'^{-1}(x)}$ are linear for all $x \in X$. Therefore, the vector bundles π and π' are isomorphic. ∎

Definition 3.3.4. *Let $\pi : E \to X$ be a vector bundle of rank r. A* section *of $\pi : E \to X$ is a morphism of varieties $s : X \to E$ such that $\pi \circ s = \mathrm{id}_X$. If $\pi : E \to X$ is determined by the transition functions $g_{i,j}$, then s is determined by a collection of morphisms $s_i : U_i \to \mathbb{A}^r$ such that $s_i = g_{i,j}s_j$ on $U_i \cap U_j$.*

We define several operations on vector bundles, following [36, Ch. VI]. Suppose $\pi_1 : E_1 \to X$ and $\pi_2 : E_2 \to X$ are vector bundles over X, of ranks r_1 and r_2, respectively, with transition functions $g^1_{i,j}$ and $g^2_{i,j}$, respectively.

- The *tensor product* of vector bundles $E \otimes E'$ is a vector bundle of rank $r_1 r_2$, defined by the transition functions $g^1_{i,j} \otimes g^2_{i,j} \in \mathrm{GL}(r_1 r_2, k)$.
- Suppose there is a morphism of vector bundles $\phi : E' \to E$, which is a closed embedding. Then $\phi(E')$ is called a *subbundle* of E. Any subbundle of a vector bundle is locally a direct summand (See [36, pp. 60] for further details).

- Suppose E' is a subbundle of E. It is possible to define the quotient bundle E/E'. As a set,
$$E/E' = \cup_{x\in X} E_x/F_x.$$
Putting a variety structure on the set E/E' is more complicated. We refer to [36, pp. 61] for this construction.

Example 3.3.5. Let $\{U_i\}$ be the standard affine open covering of $\mathbb{P}^n$. There is a line bundle $\pi : \mathbb{P}^n \times \mathbb{A}^1 \to \mathbb{P}^n$, where π is the map

$$\pi(x, y) = x.$$

The transition functions of this line bundle are $g_{i,j} = 1$. We call $\pi : \mathbb{P}^n \times \mathbb{A}^1 \to \mathbb{P}^n$ the *trivial line bundle* over projective n-space. We denote this line bundle by $\mathcal{O}_{\mathbb{P}^n}$.

Example 3.3.6. Let $\{U_i\}$ be the standard affine open covering of $\mathbb{P}^n$. Given a point $x \in \mathbb{P}^n$, let l_x be the line in $\mathbb{A}^{n+1}$ corresponding to x. Set

$$L = \{(x, y) \mid x \in \mathbb{P}^n,\ y \in l_x\}.$$

This is a subvariety of $\mathbb{P}^n \times \mathbb{A}^{n+1}$. There is a line bundle $\pi : L \to \mathbb{P}^n$, where π is the map
$$\pi(x, y) = x.$$
There is an isomorphism of varieties

$$\phi_i : \pi^{-1}(U_i) = \{(x, y) \mid x \in U_i,\ y \in l_x\} \to U_i \times \mathbb{A}^1, \quad \phi_i(x, (y_0, \ldots, y_n)) = (x, y_i).$$

On the intersection $U_i \cap U_j$, we have

$$\phi_i \circ \phi_j^{-1}(x, y_j) = (x, y_i).$$

Thus, the transition functions for this line bundle are $g_{i,j} = \frac{y_i}{y_j}$. We call this line bundle the *tautological line bundle* over projective n-space. We denote this line bundle by $\mathcal{O}_{\mathbb{P}^n}(-1)$.

Example 3.3.7. Let $\{U_i\}$ be the standard affine open covering of $\mathbb{P}^n$. Given a point $x \in \mathbb{P}^n$, let l_x be the line in $\mathbb{A}^{n+1}$ corresponding to x, and let F_x be the set of k-linear functionals on l_x. Set
$$F = \{(x, f) \mid x \in \mathbb{P}^n,\ f \in F_x\}.$$
This is a subvariety of $\mathbb{P}^n \times (k^{n+1})^*$. Then there is a line bundle $\pi : F \to \mathbb{P}^n$, where π is the map
$$\pi(x, f) = x.$$
There is an isomorphism of varieties

$$\phi_i : \pi^{-1}(U_i) = \{(x, f) \mid x \in U_i,\ f \in F_x\} \to U_i \times \mathbb{A}^1, \quad \phi_i(x, af_{i,x}) = (x, a),$$

where $a \in k$ and $f_{i,x}(y_{i,x}) = 1$, where $y_{i,x}$ is the point in l_x whose i-th coordinate is 1. On the intersection $U_i \cap U_j$, we have

$$\phi_i \circ \phi_j^{-1}(x, a) = \left(x, \frac{y_j}{y_i} a\right).$$

Thus, the transition functions for this line bundle are $g_{i,j} = \frac{y_j}{y_i}$. We call this line bundle the *hyperplane bundle* over projective n-space. We denote this line bundle by $\mathcal{O}_{\mathbb{P}^n}(1)$.

The hyperplane bundle over projective n-space is dual to the tautological line bundle over projective n-space in the sense that $\mathcal{O}_{\mathbb{P}^n}(-1) \otimes \mathcal{O}_{\mathbb{P}^n}(1) \simeq \mathcal{O}_{\mathbb{P}^n}$. This isomorphism holds by Lemma 3.3.3, and by the definition of the tensor product of vector bundles. Similarly, we can define a line bundle over projective n-space for any integer d:

$$\mathcal{O}_{\mathbb{P}^n}(d) := \begin{cases} \mathcal{O}_{\mathbb{P}^n}(1)^{\otimes d}, & \text{if } d > 0, \\ \mathcal{O}_{\mathbb{P}^n}, & \text{if } d = 0, \\ \mathcal{O}_{\mathbb{P}^n}(-1)^{\otimes(-d)}, & \text{if } d < 0. \end{cases}$$

Any line bundle over projective n-space is isomorphic to a line bundle the form $\mathcal{O}_{\mathbb{P}^n}(d)$ (See [18, Ch. II, Cor. 6.17]). The set of isomorphisms classes of line bundles over $\mathbb{P}^n$ forms a group with the tensor product operation. This group is called the *Picard group* of $\mathbb{P}^n$. More generally, if X is any algebraic variety over k, then the set of isomorphism classes of line bundles over X forms a group under the tensor product operation. This group is called the *Picard group of X*. We denote the Picard group of X by $\mathrm{Pic}(X)$.

Definition 3.3.8. *Let $\chi : E \to X$ be a vector bundle of rank r, and let $\{U_i\}$ be an open cover of X with transition functions $g_{i,j}$. We identify $\mathbb{P}^{r-1}$ with $(\mathbb{A}^r \setminus \{0\})/k^*$, and we define the* associated projective bundle $\mathbb{P}(E)$ *as the variety consisting of open sets $U_i \times \mathbb{P}^{r-1}$ glued along the isomorphisms*

$$(U_i \cap U_j) \times \mathbb{P}^{r-1} \to (U_i \cap U_j) \times \mathbb{P}^{r-1}, \quad (u, v) \mapsto (u, (g_{i,j}(u))(v)),$$

for all $i \neq j$. Note that $g_{i,j}(u) : \mathbb{P}^{r-1} \to \mathbb{P}^{r-1}$ is a well-defined map because the $g_{i,j}(u)$ are k-linear. There is a canonical projection map $\mathbb{P}(E) \to E$ sending $(x, P) \mapsto x$.

Remark 3.3.9. Let $\pi : E \to X$ be a vector bundle of rank n. For every line bundle $\mathcal{O}_{\mathbb{P}^{n-1}}(d) \to \mathbb{P}^{n-1}$ over projective $(n-1)$-space, there is a line bundle $\mathcal{O}_{\mathbb{P}(E)}(d) \to \mathbb{P}(E)$ over $\mathbb{P}(E)$. The line bundle $\mathcal{O}_{\mathbb{P}(E)}(1)$ is called the *canonical bundle* over $\mathbb{P}(E)$.

3.4 Chow theory

In this section, we fix once and for all a *smooth* algebraic variety X over k. Furthermore, following the convention in Fulton [15, Ch. 1, §1.7], we will always assume in this section that a flat morphism of varieties has relative dimension n for some integer n. The goal of this section is to introduce the Chow theory of X. We closely follow the expositions in Atiyah-Macdonald [1], Fulton [15], Hartshorne [18], and Smith [37].

Following [1, §9], we introduce the concept of a *discrete valuation ring* (DVR) below.

Definition 3.4.1. *Let K be the field of fractions of an integral domain R. A* discrete valuation *on K is a function $\nu : K^\times \to \mathbb{Z}$ such that*

$$\nu(xy) = \nu(x) + \nu(y) \quad \textit{and} \quad \nu(x+y) \geq \min\{\nu(x), \nu(y)\},$$

for all $x, y \in K$. If

$$R = \{0\} \cup \{x \in K \mid \nu(x) \geq 0\},$$

then R is called a discrete valuation ring (DVR)*, with valuation ν.*

By [1, Prop. 9.6], a DVR equivalently can be defined as follows:

Definition 3.4.2. *An integral domain R is a* discrete valuation ring (DVR) *if it is not a field, and if it is a Noetherian local domain whose maximal ideal is principal.*

The discussion from here up to Lemma 3.4.3 closely follows [37, §17].

A *prime divisor* of X is an irreducible closed subvariety of codimension 1 in X. Let D be a prime divisor of X. First assume X is affine. By [37, §17], the maximal ideal of $\mathcal{O}_{X,D}$ is principal, and $\mathcal{O}_{X,D}$ is a Noetherian local domain. Therefore, $\mathcal{O}_{X,D}$ is a DVR. Let $k(X)^\times$ be the group of units of the function field $k(X)$ of X, and let t be the generator of the maximal ideal of $\mathcal{O}_{X,D}$. Given an element $\langle U, f\rangle \in k(X)^\times$, there are polynomials $g, h \in k[X]$ such that $f = \frac{g}{h}$ and such that h does not vanish on U. Thus, $g, h \in \mathcal{O}_{X,D}$, so we can write $g = u_1 t^{n_1}$ and $h = u_2 t^{n_2}$ for some units u_1 and u_2. Therefore, $f = (u_1 u_2^{-1}) t^{n_1 - n_2}$. There is a group homomorphism

$$\operatorname{ord}_D : k(X)^\times \to \mathbb{Z}, \quad f = ut^n \mapsto n.$$

The function ord_D is called the *order of vanishing* of f along D. It is a valuation function for $\mathcal{O}_{X,D}$ in $k(X)$.

Now assume X is any smooth algebraic variety over k. We can choose a smooth open affine subset U of X such that $U \cap D$ is nonempty and $U \cap D$ is a prime divisor in U. Note that $k(X) = k(U)$ because nonempty open sets are dense in the Zariski topology. Given $f \in k(X)^\times = k(U)^\times$, define $\operatorname{ord}_D^U(f)$ as before. By [37, Claim 17.15], this integer is independent of U, so we set $\operatorname{ord}_D(f) := \operatorname{ord}_D^U(f)$. This is the *order of vanishing* of f along D.

Suppose $\langle U, f\rangle \in k(X)^\times$, and set

$$\operatorname{div}(f) = \sum_D \operatorname{ord}_D(f) D,$$

where the sum runs over all prime divisors D of X.

Lemma 3.4.3. *The sum* $\operatorname{div}(f) = \sum_D \operatorname{ord}_D(f)D$ *is finite.*

Proof: First assume X is affine. Write $f = \frac{g}{h}$, where $g, h \in k[X]$ and h does not vanish on U. Since X is Noetherian, any closed subset of X contains finitely many irreducible components. The result then follows in the affine case because there are finitely many prime divisors contained in the zero locus of (g) and in the zero locus of (h). Now the result follows for any smooth algebraic variety X over k, since X admits a finite affine open covering. ∎

The discussion from here up to Lemma 3.4.4 is taken out of [15, Ch. 1, Ch. 8].

An r-*cycle* in X is a finite formal linear combination

$$\sum n_i[V_i],$$

where $n_i \in \mathbb{Z}$, and the V_i are r-dimensional subvarieties of X. Let Z_r be the group of r-cycles on X. Given an $(r+1)$-dimensional subvariety W of X and a rational function $f \in k(W)^\times$, we define an r-cycle $[\operatorname{div}(f)]$ on X by

$$[\operatorname{div}(f)] = \sum \operatorname{ord}_V(f)[V],$$

where the sum is taken over all codimension 1 subvarieties V of W. An r-cycle α is *rationally equivalent to zero* if there are finitely many $(r+1)$-dimensional subvarieties W_i of X, and rational functions $f_i \in k(W_i)^\times$, such that

$$\alpha = \sum [\operatorname{div}(f_i)].$$

As $[\operatorname{div}(f^{-1})] = -[\operatorname{div}(f)]$, the r-cycles rationally equivalent to zero form a subgroup B_r of Z_r.

Set $\mathrm{CH}_r(X) := Z_r/B_r$. We call $\mathrm{CH}_*(X) := \bigoplus_{i \geq 0} \mathrm{CH}_i(X)$ the *Chow group* of X. The subgroup $\mathrm{Cl}(X) := \mathrm{CH}_{\dim(X)-1}(X)$ of $\mathrm{CH}_*(X)$ is called the *divisor class group* of X. Now set $\mathrm{CH}^r(X) := \mathrm{CH}_{\dim(X)-r}(X)$. Since X is smooth, there is a product on $\mathrm{CH}^*(X)$ called the *intersection product* turning $\mathrm{CH}^*(X)$ into a graded, commutative, unital ring, with unit $[X]$. With this additional product structure, we call the ring $\mathrm{CH}^*(X)$ the *Chow ring* of X. In fact, we can view CH^* as a contravariant functor from the category of smooth algebraic varieties over k to the category of graded commutative unital rings. This is discussed further in Section 3.5. The functor CH^* is called the *Chow theory*.

We now state several functorial properties of the Chow theory.

(1) *Flat pull-backs.*

Let $f : X \to Y$ be a flat morphism of algebraic varieties over k of relative dimension r. For any subvariety V of Y, set

$$f^*[V] = [f^{-1}(V)].$$

The map f^* extends, by linearity, to a ring homomorphism $f^* : \mathrm{CH}^*(Y) \to \mathrm{CH}^*(X)$.

(2) *Proper push-forwards.*

Let $f : X \to Y$ be a proper morphism of algebraic varieties over k. Let Z be a subvariety of X, and set

$$f_*([Z]) = \begin{cases} 0, & \text{if } \dim f(Z) < \dim Z, \\ [k(Z) : k(f(Z))][f(Z)], & \text{if } \dim f(Z) = \dim Z. \end{cases}$$

Here, $[k(Z) : k(f(Z))]$ denotes the degree of the finite field extension $k(Z)/k(f(Z))$ (The reason this is a finite field extension is explained in [15, Appendix B.2.2]). The map f_* extends to a group homomorphism $f_* : \mathrm{CH}_*(X) \to \mathrm{CH}_*(Y)$.

(3) *Localization exact sequence.*

Suppose Y is a closed subvariety of X, and let $U = X \setminus Y$ be the open complement of Y in X. Let $i : Y \to X$ and $j : U = X \setminus Y \to X$ be the inclusion maps. The following sequence is exact:

$$\mathrm{CH}^*(Y) \xrightarrow{i_*} \mathrm{CH}^*(X) \xrightarrow{j^*} \mathrm{CH}^*(U) \longrightarrow 0 .$$

(4) *Pullback squares.*

Let

$$\begin{array}{ccc} X' & \xrightarrow{g'} & X \\ {\scriptstyle f'}\downarrow & & \downarrow{\scriptstyle f} \\ Y' & \xrightarrow[g]{} & Y \end{array}$$

be a pullback square (in the sense of Definition 2.1.12), with g flat and f proper. Then g' is flat, f' is proper, and $f'_* g'^* = g^* f_*$.

(5) *Homotopy invariance.* (See Fulton [15, §3.3].)

Let $\pi : E \to X$ be a vector bundle of rank r. The morphism π is flat, and it induces an isomorphism

$$\pi^* : \mathrm{CH}_{n-r}(X) \to \mathrm{CH}_n(E), \quad \text{for all } n.$$

Lemma 3.4.4. *There is a natural isomorphism* $\mathrm{Cl}(X) \simeq \mathrm{Pic}(X)$.

Proof: See [18, Ch. II, Cor. 6.16]. ∎

The rest of this section closely follows [18, Appendix A.2, A.3].

Let $\pi : E \to X$ be a vector bundle of rank r over X, with associated projective bundle $p : \mathbb{P}(E) \to X$. Let $\zeta \in \mathrm{CH}^1(\mathbb{P}(E)) = \mathrm{Cl}(\mathbb{P}(E))$ be the element corresponding to the class of the canonical bundle $[\mathrm{O}_{\mathbb{P}(E)}(1)] \in \mathrm{Pic}(\mathbb{P}(E))$. Through the flat pullback $\pi^* : \mathrm{CH}^*(X) \to \mathrm{CH}^*(\mathbb{P}(E))$, the ring $\mathrm{CH}^*(\mathbb{P}(E))$ is a free $\mathrm{CH}^*(X)$-module with basis $\{1, \zeta, \ldots, \zeta^{r-1}\}$. For each $i = 0, \ldots, r$, we define the *i-th Chern class* $c_i(E) \in \mathrm{CH}^i(X)$ by the requirements $c_0(E) = 1$, and

$$\sum_{i=0}^{r} (-1)^i \pi^*(c_i(E)) \cdot \zeta^{r-i} = 0.$$

3.5 Oriented cohomology theories

In this section we define the oriented cohomology theory. We closely follow [31].

Let $\mathcal{V}$ be the category whose objects are smooth algebraic varieties over k, and whose morphisms are morphisms of algebraic varieties over k. Let $\mathcal{R}^*$ be the category of graded commutative unital rings.

Remark 3.5.1. We are almost ready to define the oriented cohomology theory. The definition will use the terms *projective morphism of varieties*; *tor-independent morphism of varieties*; *torsors*; *additive functor*; and *relative codimension*. These concepts are beyond the scope of this thesis, but we provide the following remarks:

- A *projective morphism* of algebraic varieties generalizes the notion of a proper morphism of algebraic varieties. A reference for projective morphisms is [18, Ch. II].
- A *tor-independent morphism* of algebraic varieties generalizes the notion of a flat morphism of algebraic varieties. A reference for tor-independent morphisms is [31, Ch. 1, Def. 1.1.1].
- The notion of a *torsor* is similar to the notion of an vector bundle. A reference for torsors is [27].
- An *additive functor* is defined in [31, pp. 2].
- The *relative codimension* of a morphism of varieties is related to the notion of relative dimension. The precise definition of relative codimension is given in [31, Ch. 1, §1.1].

Definition 3.5.2. *An* oriented cohomology theory *is an additive, contravariant functor* $\mathrm{h}^* : \mathcal{V} \to \mathcal{R}^*$ *satisfying the following axioms. For any morphism* $f : X \to Y$ *in* $\mathcal{V}$*, the pullback* $\mathrm{h}^*(f) : h^i(Y) \to \mathrm{h}^i(X)$ *will be denoted* f^*.

(1) *Given a projective morphism $f : Y \to X$ in $\mathcal{V}$ of relative codimension r, there is a homomorphism of graded $R^*(X)$-modules*

$$f_* : \mathrm{h}^n(Y) \to \mathrm{h}^{n+r}(X)$$

called the push-forward*, where the $\mathrm{h}^*(X)$-module structure on $\mathrm{h}^*(Y)$ is induced by the pullback f^*.*

(2) *We have $(\mathrm{Id}_X)_* = \mathrm{Id}_{\mathrm{h}^*(X)}$ for any $X \in \mathcal{V}$. Furthermore, given projective morphisms $f : Y \to X$ and $g : Z \to Y$, with f of relative codimension r and g of relative codimension s, we have*

$$(f \circ g)_* = f_* \circ g_* : \mathrm{h}^n(Z) \to \mathrm{h}^{n+r+s}(X).$$

(3) *Let $f : X \to Z$ and $g : Y \to Z$ be tor-independent morphisms in $\mathcal{V}$, giving a pullback diagram*

$$\begin{array}{ccc} W & \xrightarrow{g'} & X \\ {\scriptstyle f'}\downarrow & & \downarrow{\scriptstyle f} \\ Y & \xrightarrow[g]{} & Z \end{array} .$$

If f is projective of relative dimension r, then $g^* f_* = f'_* g'^*$.

(4) Projective bundle formula*: Let $E \to X$ be a vector bundle of rank n over $X \in \mathcal{V}$. Let $O_{\mathbb{P}(E)}(1) \to \mathbb{P}(E)$ be the canonical bundle, with zero section $s : \mathbb{P}(E) \to \mathcal{O}_{\mathbb{P}(E)}(1)$. Let $1 \in \mathrm{h}^0(\mathbb{P}(E))$ be the multiplicative unit element. Set $\zeta := s^*(s_*(1)) \in \mathrm{h}^1(\mathbb{P}(E))$. Then $\mathrm{h}^*(\mathbb{P}(E))$ is a free $\mathrm{h}^*(X)$-module, with basis*

$$\{1, \zeta, \dots, \zeta^{n-1}\}.$$

(5) Extended homotopy property*: Let $E \to X$ be a vector bundle over some $X \in \mathcal{V}$, and let $p : V \to X$ be an E-torsor. Then $p^* : \mathrm{h}^*(X) \to \mathrm{h}^*(V)$ is an isomorphism.*

A morphism *of oriented cohomology theories* $\mathrm{h} \to \mathrm{h}'^*$ *is a natural transformation* $\eta : \mathrm{h}^* \to \mathrm{h}'^*$ *that commutes with the pushforwards.*

Let h^* be an oriented cohomology theory, and let $X \in \mathcal{V}$. Given a vector bundle $E \to X$ of rank n, there exist unique elements $c_i(E) \in \mathrm{h}^i(X)$, $i \in \{0, \dots, n\}$, called *Chern classes*, such that $c_0(E) = 1$ and

$$\sum_{i=0}^{n} (-1)^i c_i(E) \zeta^{n-i} = 0.$$

The Chern classes satisfy:

(1) Given a line bundle L over $X \in \mathcal{V}$, $c_1(L)$ equals $s^* s_*(1) \in \mathrm{h}^1(X)$, where $s\colon X \to L$ is the zero section.

(2) Given a morphism $f : Y \to X$ in $\mathcal{V}$, and a vector bundle E over X, we have

$$c_i(f^*(E)) = f^*(c_i(E)), \quad i \geq 0.$$

(3) *The Whitney product formula.* If

$$0 \to E' \to E \to E'' \to 0$$

is an exact sequence of vector bundles, then for all $n \geq 0$, we have

$$c_n(E) = \sum_{i=1}^{n} c_i(E') c_{n-i}(E'').$$

Furthermore, the Chern classes satisfy *Quillen's formula*. Let L_1 and L_2 be two line bundles over X. Then

$$c_1(L_1 \otimes L_2) = c_1(L_1) +_F c_1(L_2),$$

where F is a one-dimensional commutative formal group law over the coefficient ring $R = \mathrm{h}^*(\mathrm{pt})$, where pt is any point in X. In particular, to each oriented cohomology theory h^* we can associate a formal group law F. The following theorem holds because k has characteristic 0:

Theorem 3.5.3. *There exists an oriented cohomology theory Ω^* called* algebraic cobordism *such that, for any oriented cohomology theory* h^**, there is a unique morphism of oriented cohomology theories $\Omega^* \to \mathrm{h}^*$. The formal group law corresponding to Ω^* is the universal formal group law F_U over the Lazard ring $\mathbb{L}$.*

Proof: See [31, Thm. 1.2.6]. ∎

Example 3.5.4. (See [31, Example 1.1.4].) The Chow theory CH^* is an example of an oriented cohomology theory. It corresponds to the additive formal group law F_A over the coefficient ring $R = \mathbb{Z}$.

Example 3.5.5. (See [31, Example 1.1.5] and [43].) Let X be a smooth algebraic variety over k. Set $\mathrm{K}^0(X) := F(X)/C(X)$, where $F(X)$ is the free abelian group generated by isomorphism classes of vector bundles over X, and $C(X)$ is the subgroup of $F(X)$ generated by elements of the form $[E] - [E'] - [E'']$ whenever there is a short exact sequence of vector bundles

$$0 \to E' \to E \to E'' \to 0$$

over X. We call $\mathrm{K}^0(X)$ the *Grothendieck group of vector bundles* over X. The tensor product of vector bundles induces a product structure on $\mathrm{K}^0(X)$, turning it into a graded, commutative, unital ring. If β is an indeterminate, then the assignment $X \mapsto \mathrm{K}^0(X)[\beta, \beta^{-1}] := \mathrm{K}^0(X) \otimes_{\mathbb{Z}} \mathbb{Z}[\beta, \beta^{-1}]$ defines an oriented cohomology theory. The formal group law associated with $\mathrm{K}^0(-)[\beta, \beta^{-1}]$ is the multiplicative formal group law F_M with respect to β over the coefficient ring $R = \mathbb{Z}[\beta, \beta^{-1}]$. The element β is known as the *Bott element*.

Example 3.5.6. Given a formal group law F over a commutative, unital ring R, there is an oriented cohomology theory

$$\mathrm{h}_F^*(-) = \Omega^*(-) \otimes_{\mathbb{L}} R,$$

where R is an $\mathbb{L}$-module via the morphism $\mathbb{L} \to R$ of Proposition 3.1.6. A theory constructed in this way is called a *free oriented cohomology theory*.

Remark 3.5.7. The Chow theory CH^* and the functor $\mathrm{K}^0(-)[\beta, \beta^{-1}]$ are free oriented cohomology theories (See [31, Theorem 1.2.18 and Theorem 1.2.19]).

Chapter 4

Linear algebraic groups

In this chapter, we fix once and for all an algebraically closed field k of characteristic 0. The purpose of this chapter is to describe linear algebraic groups over k. In Section 4.1, we introduce root systems and Dynkin diagrams. In Section 4.2, we introduce the weight lattice and root datum. In Section 4.3, we introduce linear algebraic groups over k. In particular, we summarize the classification of connected reductive groups over k in terms of root data. In Section 4.4, we introduce the complete flag variety and discuss line bundles over complete flag varieties. The main references we use in this chapter are Björner-Brenti [2], Borel [3], Bourbaki [6], Humphreys [21], [22], and [23], and Springer [39].

4.1 Root systems and Dynkin diagrams

In this section, we introduce roots systems, Dynkin diagrams, and Weyl groups. We closely follow Humphreys [21], [22], and [23], and Björner-Brenti [2].

Let V be a real finite dimensional vector space, and let $(\cdot,\cdot)$ be the standard inner product on V. For any nonzero vector $\alpha \in V$, there is a linear operator s_α on V defined by the formula

$$s_\alpha(\lambda) = \lambda - 2\frac{(\alpha,\lambda)}{(\alpha,\alpha)}\alpha, \quad \lambda \in V.$$

The operator s_α has order 2, and it is called the *reflection across* α.

Definition 4.1.1. *A* root system *in* V *is a subset* Σ *of* V*, satisfying the following axioms:*

(1) Σ *is finite, non-empty, spans* V*, and does not contain* 0.

(2) If $\alpha \in \Sigma$*, then the only multiples of* α *in* Σ *are* $\pm\alpha$.

(3) If $\alpha \in \Sigma$, then the reflection s_α stabilizes Σ.

(4) If $\alpha, \beta \in \Sigma$, then $s_\alpha(\beta) - \beta = n\alpha$ for some $n \in \mathbb{Z}$.

The elements in Σ are called roots.

Let Σ and Σ' be two root systems, which generate the vector spaces V and V', respectively. A morphism $\phi : \Sigma' \to \Sigma$ is a morphism of vector spaces $\phi : V' \to V$, which maps Σ' to Σ and preserves the integers n of axiom (4) of Definition 4.1.1.

For the rest of the section, we fix a root system Σ of V. The *rank* of Σ is the dimension of V. The root system Σ is *irreducible* if it cannot be partitioned into the union of two mutually orthogonal proper subsets.

Definition 4.1.2. *The group W generated by the reflections s_α, over all $\alpha \in \Sigma$, is called the* Weyl group *of Σ.*

Given a root α, the *coroot* $\alpha^\vee : V \to \mathbb{R}$ corresponding to α is the linear functional defined by the formula

$$\alpha^\vee(\lambda) = 2\frac{(\alpha, \lambda)}{(\alpha, \alpha)}, \quad \lambda \in V.$$

Thus, $s_\alpha(\lambda) = \lambda - \alpha^\vee(\lambda)\alpha$ for all $\alpha \in \Sigma$ and $\lambda \in V$. The set of coroots $\Sigma^\vee$ forms a root system called the *root system dual to* Σ. There is a bijection $\Sigma \to \Sigma^\vee$, given by $\alpha \mapsto \alpha^\vee$. Given any two roots α and β, we compute the possible angles between α and β. Since $(\alpha, \beta) = |\alpha||\beta|\cos(\theta)$, axiom (4) of Definition 4.1.1 implies that

$$\alpha^\vee(\beta)\beta^\vee(\alpha) = (2\cos(\theta))^2$$

is an integer. Thus, $\theta \in \{0°, 30°, 45°, 60°, 90°, 120°, 135°, 150°, 180°\}$. Therefore, if $m_{\alpha,\beta}$ is the order of $s_\alpha s_\beta$ in W, then, by direct computation, one sees that $m_{\alpha,\beta} \in \{1, 2, 3, 4, 6\}$. Observe that $m_{\alpha,\beta} = 1$ if and only if $\theta \in \{0°, 180°\}$. By axiom (2) of Definition 4.1.1, $\theta \in \{0°, 180°\}$ if and only if $\beta = \pm\alpha$. If Σ is irreducible, then, by [21, §10.4, Lemma C] there are at most two lengths of roots that occur in Σ. Moreover, all roots of the same length are conjugate in Σ under W.

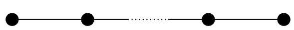
We call a subset $\Delta = \{\alpha_1, \ldots, \alpha_n\}$ of Σ a *simple system* of Σ if it is a vector space basis of V, and if any $\beta \in \Sigma$ is expressible as a linear combination $\beta = \sum_{\alpha\in\Delta} c_\alpha \alpha$, where the $c_\alpha \in \mathbb{R}$ are all non-negative (in which case β is called a *positive root*) or all non-positive (in which case β is called a *negative root*). There is a partition $\Sigma = \Sigma^+ \cup \Sigma^-$, where Σ^+ (resp. Σ^-) consists of the positive (resp. negative) roots of Σ. The elements of Δ are called *simple roots* with respect to Σ, and the reflection along a simple root is called a *simple reflection*. By [23, Ch. I, Cor. 1.3], if $\alpha_i \neq \alpha_j$ are simple roots, then $(\alpha_i, \alpha_j) \leq 0$. Thus, the angle between α_i and α_j is $\theta \in \{90°, 120°, 135°, 150°\}$. The *Cartan matrix* of Σ is the matrix $C_{i,j} := (\alpha_j^\vee(\alpha_i))_{i,j=1}^n$, where n is the rank of Σ.

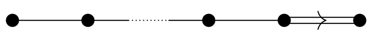
Definition 4.1.3. (see [22, Appendix A.7]) *The* Dynkin diagram *of* Δ *is a graph with a vertex for each simple root. There is an edge between the vertices* α_i *and* α_j *if and only if* $(\alpha_i, \alpha_j) < 0$*, and the multiplicity of the edge between* α_i *and* α_j *is*

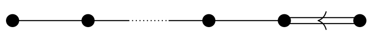
$$\operatorname{mult}(\alpha_i, \alpha_j) = \begin{cases} 1 & \textit{if } \theta = 120^\circ; \\ 2 & \textit{if } \theta = 135^\circ; \\ 3 & \textit{if } \theta = 150^\circ. \end{cases}$$

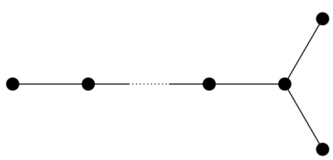
An edge in the Dynkin diagram is directed if and only if the vertices of the edges differ in length. Any directed edge points towards the shorter root.

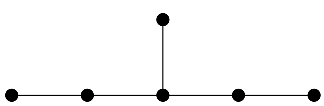
Up to isomorphism, the irreducible root systems are in one-to-one correspondence with the following Dynkin diagrams:

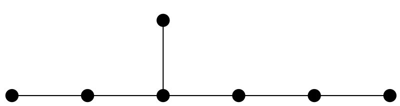
A_n 

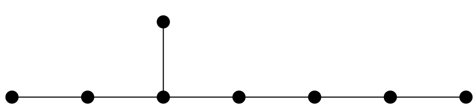
B_n 

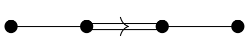
C_n 

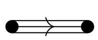
D_n 

E_6 

E_7 

E_8 

F_4 

G_2 

For $n \geq 1$, let $\{e_i\}_{i=1}^n$ be the standard basis of $\mathbb{R}^n$. We construct a root system for each Dynkin type A-G. These constructions are taken out of [21, §12.1].

Example 4.1.4. Let $n \geq 2$. The set

$$\Sigma = \{e_i - e_j \mid i \neq j\}$$

is the root system of type A_{n-1}, $n \geq 2$, and Σ has rank $n-1$. A simple system for Σ is the set

$$\Delta = \{e_i - e_{i+1} \mid 1 \leq i \leq n-1\}.$$

The Weyl group of Σ is $W = S_n$, the symmetric group on n letters.

Example 4.1.5. Let $n \geq 2$. The set

$$\Sigma = \{\pm e_i \pm e_j \mid i < j\} \cup \{\pm e_i \mid 1 \leq i \leq n\}$$

is the root system of type B_n, $n \geq 1$, and Σ has rank n. A simple system for Σ is the set

$$\Delta = \{e_i - e_{i+1} \mid 1 \leq i \leq n-1\} \cup \{e_n\}.$$

The Weyl group of Σ is the semidirect product $W = S_n \ltimes (\mathbb{Z}/2\mathbb{Z})^n$, where the i-th $\mathbb{Z}/2\mathbb{Z}$ factor acts on Σ by switching the sign of e_i.

Example 4.1.6. Let $n \geq 3$. The set

$$\Sigma = \{\pm e_i \pm e_j \mid i \neq j\} \cup \{\pm 2e_i \mid 1 \leq i \leq n\}$$

is the root system of type C_n, and Σ has rank n. A simple system for Σ is the set

$$\Delta = \{e_i - e_{i+1} \mid 1 \leq i \leq n-1\} \cup \{2e_n\}.$$

The Weyl group of Σ is the semidirect product $W = S_n \ltimes (\mathbb{Z}/2\mathbb{Z})^n$, where the i-th $\mathbb{Z}/2\mathbb{Z}$ factor acts on Σ by switching the sign of e_i.

Example 4.1.7. Let $n \geq 4$. The set

$$\Sigma = \{\pm e_i \pm e_j \mid i \neq j\}$$

is the root system of type D_n, and Σ has rank n. A simple system for Σ is the set

$$\Delta = \{e_i - e_{i+1} \mid 1 \leq i \leq n-1\} \cup \{e_{n-1} + e_n\}.$$

The Weyl group of Σ is a subgroup of the semidirect product $W = S_n \ltimes (\mathbb{Z}/2\mathbb{Z})^n$, where the i-th $\mathbb{Z}/2\mathbb{Z}$ factor acts on Σ by switching the sign of e_i. The subgroup consists of permutations and sign changes that involve an *even* number of sign changes.

Example 4.1.8. Let $n = 8$. The set

$$\Sigma = \{\pm e_i \pm e_j \mid i, j = 1, \ldots, 8\} \cup \left\{ \frac{1}{2} \sum_{i=1}^{8} (-1)^{k(i)} e_i \mid k(i) \in \{0, 1\} \text{ and } \sum_{i=1}^{8} k(i) \in 2\mathbb{Z} \right\}$$

is the root system of type E_8, and Σ has rank 8. A simple system for Σ is the set

$$\Delta = \left\{ \frac{1}{2}(e_1 + e_8 - (e_2 + \cdots e_7)), e_1 + e_2 \right\} \cup \{e_i - e_{i-1} \mid i = 2, \ldots, 7\}.$$

The Weyl group of Σ has order $2^{14}3^5 7$.
The root systems of type E_6 and E_7 are sub-root-systems of E_8.

Example 4.1.9. Let $n = 4$. The set

$$\Sigma = \{\pm e_i \mid i = 1, \ldots, 4\} \cup \{\pm e_i \pm e_j \mid i, j = 1, \ldots, 4\} \cup \left\{ \pm \frac{1}{2}(e_1 \pm e_2 \pm e_3 \pm e_4) \right\}$$

is the root system of type F_4, and Σ has rank 4. A simple system for Σ is the set

$$\Delta = \left\{ e_2 - e_3, e_3 - e_4, e_4, \frac{1}{2}(e_1 - e_2 - e_3 - e_4) \right\}.$$

The Weyl group of Σ has order 1152.

Example 4.1.10. Let $n = 3$. The set

$$\Sigma = \pm \{e_1 - e_2, e_2 - e_3, e_1 - e_3, 2e_1 - e_2 - e_3, 2e_2 - e_1 - e_3, 2e_3 - e_1 - e_2\}$$

is the root system of type G_2, and Σ has rank 2. A simple system for Σ is the set

$$\Delta = \{e_1 - e_2, -2e_1 + e_2 + e_3\}.$$

The Weyl group of Σ has order 12.

For the remainder of this section, we will discuss the Weyl group W. We closely follow Björner-Brenti [2].

A *Coxeter group* is a group G with the following presentation in terms of generators and relations:

$$G = \langle g_1, \ldots, g_t \mid (g_i g_j)^{m_{i,j}} = 1 \rangle,$$

where $m_{i,j}$ is the order of $g_i g_j$ in G (if the order exists), $m_{i,i} = 1$, $m_{i,j} \geq 2$ for $i \neq j$, and $m_{i,j} = \infty$ if there is no relation of the form $(g_i g_j)^m = 1$ for any $m \geq 1$. Let W be the Weyl group of a root system Σ, and fix a set of simple reflections $S = \{s_1, \ldots, s_n\}$ of W corresponding to a set of simple roots $\Delta = \{\alpha_1, \ldots, \alpha_n\}$. Then W is a finite

Coxeter group, with generating set S (see [2, Example 1.2.10]). If $m_{i,j}$ is the order of $s_i s_j$ in W, then recall that $m_{i,j} \in \{1, 2, 3, 4, 6\}$.

Write $w \in W$ as a product $w = s_{i_1} \cdots s_{i_r}$ of simple reflections. The *length* $l(w)$ of w is the minimal r for which such an expression exists, and we call such an expression a *reduced decomposition of* w. The *longest word* in W is the unique word in W of maximal length. Write $w' \to w$ if $l(w) > l(w')$ and $w = w'u$ for some $u \in W$. We define the *Bruhat order* on W to be the partial ordering $<$ on W defined by $w' < w$ if and only if there is a sequence $w' = w_0 \to w_1 \to \cdots \to w_m = w$.

4.2 Weight lattice and root datum

In this section, we describe the weights of a root system, and we define the root datum. We closely follow Humphreys [21] and [22, Appendix], and Springer [39]. In this section, unless otherwise stated, V is a real finite-dimensional vector space, Σ is a root system in V, $\Delta = \{\alpha_1, \ldots, \alpha_n\}$ is a simple system of Σ, and W is the Weyl group of Σ.

Definition 4.2.1. *A* weight *of Σ is a vector $\lambda \in V$ such that $\alpha^\vee(\lambda) \in \mathbb{Z}$ for all $\alpha \in \Sigma$.*

A *lattice* in V is a free subgroup of a real finite-dimensional vector space V whose rank equals the dimension of V. The *root lattice* Λ_r of Σ is the lattice in V generated by the roots in Σ, and the *weight lattice* Λ_w of Σ is the lattice in V generated by the weights of Σ. By axiom (4) of Definition 4.1.1, the lattice Λ_r is contained in Λ_w. By [21, pp. 68], the *fundamental group* Λ_w/Λ_r is finite. There is a basis $\{\lambda_1, \ldots, \lambda_n\}$ of Λ_w consisting of *fundamental weights*, which satisfy $\alpha_i^\vee(\lambda_j) = \delta_{i,j}$. By [22, Appendix A.9], depending on the Dynkin type of Σ, the fundamental group has the following structure:

- A_n: $\Lambda_w/\Lambda_r \simeq \mathbb{Z}/(n+1)\mathbb{Z}$.
- B_n, C_n, E_7: $\Lambda_w/\Lambda_r \simeq \mathbb{Z}/2\mathbb{Z}$.
- D_n: $\Lambda_w/\Lambda_r \simeq \begin{cases} \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}, & n \text{ even}, \\ \mathbb{Z}/4\mathbb{Z}, & n \text{ odd}. \end{cases}$
- E_6: $\Lambda_w/\Lambda_r \simeq \mathbb{Z}/3\mathbb{Z}$.
- E_8, F_4, G_2: $\Lambda_w/\Lambda_r = 0$.

Given a root $\alpha \in \Sigma$, we can write α as a linear combination $\sum_{t=1}^{n} d_{\alpha,t}\lambda_t$ for some $d_{\alpha,t} \in \mathbb{Z}$. If $\alpha = \alpha_i$ is a simple root, then we have

$$C_{j,i} = \alpha_j^\vee(\alpha_i) = \sum_{t=1}^{n} d_{\alpha_i,t}\alpha_j^\vee(\lambda_t) = d_{\alpha_i,j},$$

where $C_{i,j}$ is the (i,j)-th entry of the Cartan matrix of Σ. Thus, $\alpha_i = \sum_{j=1}^{n} C_{j,i}\lambda_j$. Moreover, one can solve the system of linear equations $\{\alpha_i = \sum_{j=1}^{n} C_{j,i}\lambda_j\}$ in order to express λ_j as a $\mathbb{Q}$-linear combination of the α_i.

If Σ has rank 2, there are exactly four root systems: $A_1 \times A_1$, A_2, B_2, and G_2.

Example 4.2.2. Let Σ be the root system of type $A_1 \times A_1$. The Cartan matrix of Σ is

$$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

Thus, $\alpha_1 = 2\lambda_1$ and $\alpha_2 = 2\lambda_2$. We can write

$$\lambda_1 = \tfrac{1}{2}\alpha_1 \quad \text{and} \quad \lambda_2 = \tfrac{1}{2}\alpha_2.$$

One can directly verify that

$$\alpha_1^\vee(\lambda_1) = 1; \quad \alpha_1^\vee(\lambda_1) = 0; \quad \alpha_2^\vee(\lambda_1) = 0; \quad \alpha_2^\vee(\lambda_2) = 1.$$

Note also that

$$s_2(\alpha_1) = \alpha_1; \quad s_1(\alpha_1) = -\alpha_1; \quad s_2s_1(\alpha_1) = -\alpha_1.$$

$$s_2(\alpha_2) = -\alpha_2; \quad s_1(\alpha_2) = \alpha_2; \quad s_2s_1(\alpha_2) = -\alpha_2.$$

Example 4.2.3. The Cartan matrix of A_2 is:

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}.$$

Thus, $\alpha_1 = 2\lambda_1 - \lambda_2$ and $\alpha_2 = -\lambda_1 + 2\lambda_2$. We can write

$$\lambda_1 = \tfrac{2}{3}\alpha_1 + \tfrac{1}{3}\alpha_2 \quad \text{and} \quad \lambda_2 = \tfrac{1}{3}\alpha_1 + \tfrac{2}{3}\alpha_2.$$

One can directly verify that

$$\alpha_1^\vee(\lambda_1) = 1; \quad \alpha_1^\vee(\lambda_1) = 0; \quad \alpha_2^\vee(\lambda_1) = 0; \quad \alpha_2^\vee(\lambda_2) = 1.$$

Note also that

$$s_1(\alpha_1) = -\alpha_1; \quad s_2(\alpha_1) = \alpha_1 + \alpha_2; \quad s_1s_2(\alpha_1) = \alpha_2;$$

$$s_2s_1(\alpha_1) = -\alpha_1 - \alpha_2; \quad s_1s_2s_1(\alpha_1) = -\alpha_2.$$

$$s_1(\alpha_2) = \alpha_1 + \alpha_2; \quad s_2(\alpha_2) = -\alpha_2; \quad s_1s_2(\alpha_2) = -\alpha_1 - \alpha_2;$$

$$s_2s_1(\alpha_2) = \alpha_1; \quad s_1s_2s_1(\alpha_2) = -\alpha_1.$$

Example 4.2.4. The Cartan matrix of B_2 is:

$$\begin{bmatrix} 2 & -2 \\ -1 & 2 \end{bmatrix}.$$

Thus, $\alpha_1 = 2\lambda_1 - \lambda_2$ and $\alpha_2 = -2\lambda_1 + 2\lambda_2$. We can write

$$\lambda_1 = \alpha_1 + \tfrac{1}{2}\alpha_2 \quad \text{and} \quad \lambda_2 = \alpha_1 + \alpha_2.$$

One can directly verify that

$$\alpha_1^\vee(\lambda_1) = 1; \quad \alpha_1^\vee(\lambda_1) = 0; \quad \alpha_2^\vee(\lambda_1) = 0; \quad \alpha_2^\vee(\lambda_2) = 1.$$

Note also that

$$s_2(\alpha_1) = 2\alpha_2 + \alpha_1; \quad s_1(\alpha_1) = -\alpha_1; \quad s_2s_1(\alpha_1) = -2\alpha_2 - \alpha_1; \quad s_1s_2(\alpha_1) = 2\alpha_2 + \alpha_1;$$

$$s_2s_1s_2(\alpha_1) = \alpha_1; \quad s_1s_2s_1(\alpha_1) = -2\alpha_2 - \alpha_1; \quad s_2s_1s_2s_1(\alpha_1) = -\alpha_1.$$

$$s_2(\alpha_2) = -\alpha_2; \quad s_1(\alpha_2) = \alpha_2 + \alpha_1; \quad s_2s_1(\alpha_2) = \alpha_2 + \alpha_1; \quad s_1s_2(\alpha_2) = -\alpha_2 - \alpha_1;$$

$$s_2s_1s_2(\alpha_2) = -\alpha_2 - \alpha_1; \quad s_1s_2s_1(\alpha_2) = \alpha_2; \quad s_2s_1s_2s_1(\alpha_2) = -\alpha_2.$$

Example 4.2.5. The Cartan matrix of G_2 is:

$$\begin{bmatrix} 2 & -1 \\ -3 & 2 \end{bmatrix}.$$

Thus, $\alpha_1 = 2\lambda_1 - 3\lambda_2$ and $\alpha_2 = -\lambda_1 + 2\lambda_2$. We can write

$$\lambda_1 = 2\alpha_1 + 3\alpha_2 \quad \text{and} \quad \lambda_2 = \alpha_1 + 2\alpha_2.$$

Since the λ_i are expressible as a $\mathbb{Z}$-linear combination of α_1 and α_2, the weight lattice and root lattice of G_2 coincide. We have

$$s_2(\alpha_1) = \alpha_2 + \alpha_1; \quad s_1(\alpha_1) = -\alpha_1; \quad s_2s_1(\alpha_1) = -\alpha_2 - \alpha_1; \quad s_1s_2(\alpha_1) = \alpha_2 + 2\alpha_1;$$

$$s_2s_1s_2(\alpha_1) = \alpha_2 + 2\alpha_1; \quad s_1s_2s_1(\alpha_1) = -\alpha_2 - 2\alpha_1; \quad s_2s_1s_2s_1(\alpha_1) = -\alpha_2 - 2\alpha_1;$$

$$s_1s_2s_1s_2(\alpha_1) = \alpha_2 + \alpha_1; \quad s_2s_1s_2s_1s_2(\alpha_1) = \alpha_1; \quad s_1s_2s_1s_2s_1(\alpha_1) = -\alpha_2 - \alpha_1;$$

$$s_2s_1s_2s_1s_2s_1(\alpha_1) = -\alpha_1.$$

$$s_2(\alpha_2) = -\alpha_2; \quad s_1(\alpha_2) = \alpha_2 + 3\alpha_1; \quad s_2s_1(\alpha_2) = 2\alpha_2 + 3\alpha_1; \quad s_1s_2(\alpha_2) = -\alpha_2 - 3\alpha_1;$$

$$s_2s_1s_2(\alpha_2) = -2\alpha_2 - 3\alpha_1; \quad s_1s_2s_1(\alpha_2) = 2\alpha_2 + 3\alpha_1; \quad s_2s_1s_2s_1(\alpha_2) = \alpha_2 + 3\alpha_1;$$

$$s_1s_2s_1s_2(\alpha_2) = -2\alpha_2 - 3\alpha_1; \quad s_2s_1s_2s_1s_2(\alpha_2) = -\alpha_2 - 3\alpha_1; \quad s_1s_2s_1s_2s_1(\alpha_2) = \alpha_2;$$

$$s_2s_1s_2s_1s_2s_1(\alpha_2) = -\alpha_2.$$

We define the root datum below. We closely follow [39, §7.4].

Definition 4.2.6. *A* root datum *is a quadruple* $\Psi = (\Lambda, \Sigma, \Lambda^\vee, \Sigma^\vee)$, *where*

(a) Λ *and* $\Lambda^\vee$ *are free finitely-generated abelian groups of the same rank, and there is a non-degenerate bilinear form* $\langle \cdot, \cdot \rangle : \Lambda \times \Lambda^\vee \to \mathbb{Z}$ *(The bilinear form is called a* perfect pairing*).*

(b) Σ *and* $\Sigma^\vee$ *are finite subsets of* Λ *and* $\Lambda^\vee$*, respectively, and there is a bijection* $\alpha \mapsto \alpha^\vee$ *of* Σ *onto* $\Sigma^\vee$*.*

For $\alpha \in \Sigma$*, we define endomorphisms* s_α *of* Λ *and* $s_\alpha^\vee$ *of* $\Lambda^\vee$ *by*

$$s_\alpha(x) = x - \langle x, \alpha^\vee \rangle \alpha, \quad s_\alpha^\vee(y) = y - \langle \alpha, y \rangle \alpha^\vee,$$

for all $x \in \Lambda$ *and* $y \in \Lambda^\vee$*. In addition to axiom (a) and axiom (b), the following axioms are imposed:*

(1) If $\alpha \in \Sigma$*, then* $\langle \alpha, \alpha^\vee \rangle = 2$*.*

(2) If $\alpha \in \Sigma$*, then* $s_\alpha(\Sigma) = \Sigma$*, and* $s_\alpha^\vee(\Sigma^\vee) = \Sigma^\vee$*.*

Let $\Psi = (\Lambda, \Sigma, \Lambda^\vee, \Sigma^\vee)$ be a root datum. The elements of Σ are call the *roots* of Ψ, and the elements of $\Sigma^\vee$ are called the *coroots* of Ψ. The sublattice Λ_r of Λ generated by the roots is called the *root lattice* of Ψ. We call the root datum Ψ *reduced* if, given $\alpha \in \Sigma$, the only multiples of α in Σ are $\pm\alpha$. If Ψ is reduced, then Σ is a root system in the vector space $V = \mathbb{R} \otimes_{\mathbb{Z}} \Lambda_r$. Set $\Lambda_{\mathbb{Q}} = \mathbb{Q} \otimes_{\mathbb{Z}} \Lambda_r$. The sublattice Λ_w of $\Lambda_{\mathbb{Q}}$ generated by all $\lambda \in \Lambda_{\mathbb{Q}}$ such that $\alpha^\vee(\lambda) \in \mathbb{Z}$ for all $\alpha \in \Sigma$ is called the *weight lattice* of Ψ. If $\Lambda = \Lambda_w$ (resp. $\Lambda = \Lambda_r$), then the root datum is called *simply connected* (resp. *adjoint*). The finite group W generated by the elements s_α, where $\alpha \in \Sigma$, is called the *Weyl group* of Ψ.

Let $\Psi = (\Lambda, \Sigma, \Lambda^\vee, \Sigma^\vee)$ and $\Psi_1 = (\Lambda_1, \Sigma_1, \Lambda_1^\vee, \Sigma_1^\vee)$ be two root data. A *root datum isomorphism* $\Psi_1 \to \Psi$ is an isomorphism of lattices $\Lambda_1 \to \Lambda$ mapping Σ_1 onto Σ and such that its dual maps $\Sigma^\vee$ onto $\Sigma_1^\vee$.

4.3 Linear algebraic groups

In this section, we introduce linear algebraic groups. In Subsection 4.3.1, we define linear algebraic groups over k and provide many examples of such groups. In Subsection 4.3.2, we discuss several classes of linear algebraic groups. In Subsection 4.3.3, we introduce the Lie algebra of a linear algebraic group. In Subsection 4.3.4, we discuss the classification of connected reductive linear algebraic groups over k in terms of root data. In this section, we closely follow Borel [3], Bourbaki [6], Humphreys [21] and [22], and Springer [39].

4.3.1 Definition and examples

In this subsection, we define the linear algebraic group, and we provide many examples of linear algebraic groups. In this subsection, we closely follow Humphreys [22] and Springer [39].

Recall that k is any algebraically closed field of characteristic 0. Suppose G is an affine algebraic variety over k, endowed with the structure of a group such that the multiplication and inversion maps,

$$\mu : G \times G \to G, \quad (x,y) \mapsto xy \qquad \text{and} \qquad \iota : G \to G, \quad x \mapsto x^{-1},$$

are morphisms of algebraic varieties. Then G is called an *algebraic group*. Let e be the identity element of the group G. A *morphism of algebraic groups* from algebraic group G to algebraic group G' is a map $\phi : G \to G'$, which is simultaneously a group homomorphism and a morphism of algebraic varieties.

Example 4.3.1. The *general linear group* $\mathrm{GL}(n,k)$ of $n \times n$ invertible matrices with entries in k can be viewed as the closed subvariety

$$\{(g,x) \in \mathrm{M}(n,k) \times \mathbb{A}^1, \mid \det(g)x - 1 = 0\} \subseteq \mathbb{A}^{n^2+1}.$$

Here, we view $\mathrm{M}(n,k)$ as the group $\mathbb{A}^{n^2}$. Hence $\mathrm{GL}(n,k)$ is an algebraic group.

A *closed* subgroup of an algebraic group is an algebraic group.

Definition 4.3.2. *Suppose G is a closed subgroup of* $\mathrm{GL}(n,k)$*, where $n \geq 1$. We call G a* linear algebraic group*.*

Theorem 4.3.3. *Let G be an algebraic group over k. Then G is isomorphic to a linear algebraic group over k.*

Proof: This is proven in [22, Theorem 8.6]. ∎

Example 4.3.4. The *additive group* $\mathbb{G}_a$ is the affine line $\mathbb{A}^1$ with the group laws $\mu(x,y) = x + y$ and $\iota(x) = -x$.

Example 4.3.5. The *multiplicative group* $\mathbb{G}_m$ is the affine open $k^* \subseteq \mathbb{A}^1$ with the group laws $\mu(x,y) = xy$ and $\iota(x) = x^{-1}$.

Example 4.3.6. Since S_n is the closed subgroup of $\mathrm{GL}(n,k)$ consisting of permutation matrices, S_n can be viewed a linear algebraic group over k. Let G be any finite group. Then G is a subgroup of the symmetric group S_n. In particular, any finite group is a linear algebraic group over k.

Example 4.3.7. Let μ_n be the multiplicative group generated by the n-th roots of unity over k. By Example 4.3.6, μ_n is a linear algebraic group over k.

Example 4.3.8. The *special linear group* $\mathrm{SL}(n,k)$ is the closed subgroup of $\mathrm{GL}(n,k)$ defined as follows:

$$\mathrm{SL}(n,k) = \{A \in \mathrm{GL}(n,k) \mid \det(A) = 1\}.$$

Example 4.3.9. The *symplectic group* $\mathrm{Sp}(2n,k)$ is the closed subgroup of $\mathrm{GL}(2n,k)$ defined as follows:

$$\mathrm{Sp}(2n,k) = \{A \in \mathrm{GL}(2n,k) \mid A^T \Omega A = \Omega\}, \quad \Omega = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}.$$

Example 4.3.10. The *orthogonal group* $\mathrm{O}(n,k)$ is the closed subgroup of $\mathrm{GL}(n,k)$ defined as follows:

$$\mathrm{O}(n,k) = \{A \in \mathrm{GL}(n,k) \mid A^T A = I_n\}.$$

Example 4.3.11. The *spinor group* $\mathrm{Spin}(n,k)$ and the *semispinor groups* $\mathrm{SSpin}(n,k)^{\pm}$ are closed subgroup of $\mathrm{GL}(n,k)$, and they are related to Clifford algebras. See [27, Ch. VI] for a detailed construction of these groups.

Proposition 4.3.12. *Let G be a linear algebraic group over k, and let N be a closed normal subgroup of G. Then the quotient group G/N is a linear algebraic group over k.*

Proof: See [39, Proposition 5.5.10]. ∎

Example 4.3.13. The *projective general linear group* $\mathrm{PGL}(n,k)$ is the quotient group $\mathrm{GL}(n,k)/Z(\mathrm{GL}(n,k))$, where $Z(\mathrm{GL}(n,k))$ is the center of $\mathrm{GL}(n,k)$. Since $Z(\mathrm{GL}(n,k))$ consists of scaled identity matrices, $Z(\mathrm{GL}(n,k))$ is a closed normal subgroup of $\mathrm{GL}(n,k)$. Thus, by Proposition 4.3.12, $\mathrm{PGL}(n,k)$ is a linear algebraic group over k. Similarly, the *projective special linear group* $\mathrm{PSL}(n,k) = \mathrm{SL}(n,k)/Z(\mathrm{SL}(n,k))$, the *projective symplectic group* $\mathrm{PSp}(2n,k) = \mathrm{Sp}(2n,k)/Z(\mathrm{Sp}(2n,k))$, and the *projective orthogonal group* $\mathrm{PO}(n,k) = \mathrm{PO}(n,k)/Z(\mathrm{PO}(n,k))$ are all linear algebraic groups over k. Note that $Z(\mathrm{SL}(n,k))$ is the set of diagonal matrices λI_n, with $\lambda \in k$ and $\lambda^n = 1$. Thus, we can identify $Z(\mathrm{SL}(n,k))$ with μ_n. More generally, if l divides n, then the quotient $\mathrm{SL}(n,k)/\mu_l$ is a linear algebraic group over k.

We closely follow [39, Ch. 17, §17.4] for the construction of the exceptional linear algebraic group G_2 below.

Example 4.3.14. A *composition algebra* C is a finite-dimensional k-algebra with an identity element, together with a non-degenerate quadratic form N such that

$$N(xy) = N(x)N(y), \quad x, y \in C.$$

If C is 8-dimensional, then C is called an *octonion algebra*. Let $\mathbf{O}$ be the octonion algebra defined as follows. As a vector space, we define

$$\mathbf{O} = \mathrm{M}(2, k) \oplus \mathrm{M}(2, k).$$

Given

$$x = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathrm{M}(2, k),$$

we set

$$\bar{x} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

The multiplication in $\mathbf{O}$ is given by

$$(x, y)(u, v) = (xu + \bar{v}y, vx + y\bar{u}),$$

and the quadratic form is given by

$$N((x, y)) = x\bar{x} - y\bar{y}.$$

The automorphism group of $\mathbf{O}$ is a linear algebraic group, and we call this group G_2.

4.3.2 Properties of linear algebraic groups

In this subsection, we describe several classes of linear algebraic groups. We closely follow Borel [3], Humphreys [22], and Springer [39].

First, suppose G is any group. There is a function $\{\cdot, \cdot\} : G \times G \to G$ called the *commutator* of G given by $\{x, y\} = xyx^{-1}y^{-1}$. If G is abelian, then the commutator of G is trivial. Given subgroups A and B of G, let $\{A, B\}$ be the *commutator subgroup* of G with respect to A and B. The group $\{A, B\}$ is generated by elements of the form $\{a, b\}$, $a \in A$, $b \in B$. The *derived series* of G is the series defined inductively by

$$D^0 G = G \quad \text{and} \quad D^{i+1} G = \{D^i G, D^i G\}, \quad i > 0.$$

A group G is *solvable* if its derived series terminates in $\{e\}$. An element r of a unital ring R is *unipotent* if $(r - 1)^n = 0$ for some $n > 0$.

Now let G be a linear algebraic group over k. By [21, Prop. 17.2], $D^i G$ is a closed normal subgroup of G. The irreducible component of G containing the identity e is called the *identity component*. We denote the identity component of G by G°. We say that G is *connected* if $G = G^\circ$. By [39, Theorem 4.3.7], a connected linear algebraic group is smooth as an algebraic variety.

By [22, Lemma 17.3] and the discussion at the start of [22, §19.5], G contains a unique maximal closed normal solvable subgroup $R(G)$. The subgroup $R(G)$ is called the *radical* of G. The subgroup $R_u(G)$ of $R(G)$ consisting of unipotent elements is called the *unipotent radical* of G. For the rest of this chapter, we follow the definitions of semisimple and reductive groups used in [39]:

Definition 4.3.15. *Let $G \neq \{e\}$ be a linear algebraic group over k. We call G* semisimple *if $R(G) = \{e\}$ and* reductive *if $R_u(G) = \{e\}$*

Remark 4.3.16. Let G be a linear algebraic group over k. The following definitions of semisimple and reductive groups are used in [22]. The group G is semisimple if $G \neq \{e\}$ is connected and $R(G) = \{e\}$. The group G is reductive if $G \neq \{e\}$ is connected and $R_u(G) = \{e\}$.

A *Borel subgroup* of G is a maximal closed connected solvable subgroup of G. By [39, Theorem 6.2.7(iii)], all Borel subgroups in G are conjugate. We call G a *torus of rank* n if it is isomorphic to a direct product $\mathbb{G}_m \times \cdots \times \mathbb{G}_m$ of n copies of $\mathbb{G}_m$ over k. We say that G is *diagonalizable* if it is isomorphic to a closed subgroup of the group of diagonal matrices in $\mathrm{GL}(n, k)$. By definition, any torus is diagonalizable. By [39, Cor. 3.2.7(ii)], any torus is connected. In addition, any torus is solvable.

Let G be a linear algebraic group over k. Being connected and solvable, any maximal torus in G lies in a Borel subgroup of G. By [39, Theorem 6.4.1], any two maximal tori in G are conjugate.

Theorem 4.3.17. *Let G be a connected reductive group over k, and let B be a Borel subgroup of G. The set of unipotent elements B_u equals the commutator subgroup $\{B, B\}$ of B, and B_u is a closed, connected, nilpotent, normal subgroup of B in G. Moreover, B/B_u is a torus. Finally, if T is any maximal torus of G sitting in B, then $B = TB_u$ (this is a semidirect product), and the restriction of the projection $B \to B/B_u$ to T defines an isomorphism $T \simeq B/B_u$.*

Proof: See [39, Cor. 6.3.3 and Thm. 6.3.5] and [3, Thm. 10.6]. ∎

Lemma 4.3.18. *Let $\phi : G \to G'$ be a surjective morphism of connected linear algebraic groups over k. If H is a Borel subgroup (resp. maximal torus, maximal connected unipotent subgroup) in G, then $\phi(H)$ is a Borel subgroup (resp. maximal torus, maximal connected unipotent subgroup) in G'.*

Proof: This follows from [22, §21.3, Cor. C]. ∎

Lemma 4.3.19. *If B is a Borel subgroup of a connected reductive group G, then $Z(G) = Z(B)$.*

Proof: See [22, §22.2, Cor. B]. ∎

Example 4.3.20. Let $G = \mathrm{GL}(n, k)$. Then G is connected. Set

$B = \{$the set of all upper triangular matrices in $G\}$;
$U = \{$the set of all upper triangular matrices in G with only 1's along the diagonal$\}$;
$T = \{$the set of all diagonal matrices in $G\}$.

Then B is a Borel subgroup of G; U is unipotent subgroup of G; and T is a maximal torus in G. The group G is *not* semisimple. However, G is reductive.

Example 4.3.21. The linear algebraic groups $\mathrm{SL}(n, k)$, $\mathrm{SO}(n, k)$, and $\mathrm{Sp}(2n, k)$ are connected and semisimple.

4.3.3 Lie algebra of a linear algebraic group

In this subsection, we define the Lie algebra of a linear algebraic group. We closely follow Bourbaki [6] and Humphreys [21] and [22].

Definition 4.3.22. *A* Lie algebra *over k is a k-vector space $\mathfrak{g}$ with a bilinear map $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ called the Lie bracket satisfying the following axioms:*

(1) $[x, x] = 0$ *for all* $x \in \mathfrak{g}$.

(2) $[x, [y, z]] + [z, [x, y]] + [y, [z, x]] = 0$ *for all* $x, y, z \in \mathfrak{g}$.

Example 4.3.23. Any associative algebra $\mathcal{A}$ over k is a Lie algebra with Lie bracket

$$[A, B] = AB - BA \quad \text{for all } A, B \in \mathcal{A}.$$

A Lie algebra $\mathfrak{g}$ is *abelian* if $[x, y] = 0$ for all $x, y \in \mathfrak{g}$. An *ideal* in $\mathfrak{g}$ is a vector subspace I of $\mathfrak{g}$ such that $[A, B] \in I$ for all $A, B \in I$.

Definition 4.3.24. *A Lie algebra $\mathfrak{g}$ is* simple *if it is non-abelian, and if its only proper ideal is* (0). *Note that the* 1*-dimensional Lie algebra with trivial Lie bracket is* not *considered a simple Lie algebra. A Lie algebra $\mathfrak{g}$ is* semisimple *if it is the direct sum of simple Lie algebras.*

The *lower central series* of a Lie algebra $\mathfrak{g}$ is the sequence

$$\mathfrak{g} \supseteq [\mathfrak{g}, \mathfrak{g}] \supseteq [\mathfrak{g}, [\mathfrak{g}, \mathfrak{g}]] \supseteq \cdots .$$

The Lie algebra $\mathfrak{g}$ is *nilpotent* if its lower central series terminates in (0). A Lie subalgebra $\mathfrak{h}$ of the Lie algebra $\mathfrak{g}$ is *self-normalizing* if, whenever $Y \in \mathfrak{g}$ and $[X, Y] \in \mathfrak{h}$ for all $X \in \mathfrak{h}$, we have $Y \in \mathfrak{h}$.

Definition 4.3.25. *A* Cartan subalgebra $\mathfrak{h}$ *of a Lie algebra* $\mathfrak{g}$ *is a Lie subalgebra of* $\mathfrak{g}$ *that is nilpotent and self-normalizing.*

Theorem 4.3.26. *Suppose* $\mathfrak{g}$ *is a semisimple Lie algebra over* k*, and that* $\mathfrak{h}$ *is a Cartan subalgebra of* $\mathfrak{g}$*. Given a nonzero functional* α *on* $\mathfrak{h}$*, set*

$$\mathfrak{g}_\alpha = \{x \in \mathfrak{g} \mid [h, x] = \alpha(h)x \text{ for all } h \in \mathfrak{h}\}.$$

There is a Cartan decomposition*:*

$$\mathfrak{g} = \mathfrak{h} \oplus \bigoplus_{\alpha \in \Sigma} \mathfrak{g}_\alpha,$$

where Σ *is the set of nonzero functionals on* $\mathfrak{h}$ *such that* $\mathfrak{g}_\alpha \neq (0)$*. The set* Σ *is a root system in the sense of Definition 4.1.1.*

Definition 4.3.27. *Let* G *be a linear algebraic group over* k*, and let* X *be an algebraic variety over* k*. We say that* G acts *on* X *if there is a morphism of algebraic varieties* $\phi\colon G \times X \to X$*, such that*

(1) $\phi(x_1, \phi(x_2, y)) = \phi(x_1 x_2, y)$ *for all* $x_1, x_2 \in G$*,* $y \in X$*.*

(2) $\phi(e, y) = y$ *for all* $y \in X$*.*

We set $x \cdot y := \phi(x, y)$ *for all* $x \in G$*,* $y \in X$*.*

Let $G \times X \to X$ be an action of a linear algebraic group G on an algebraic variety X. For $x \in G$, let $\gamma_x : X \to X$ be the morphism of algebraic varieties $\gamma_x(y) = x^{-1} \cdot y$. Then γ_x induces a ring homomorphism $\lambda_x : k[X] \to k[X]$ on coordinate rings defined by the rule $(\lambda_x(f))(y) = f(\gamma_x(y)) = f(x^{-1} \cdot y)$.

The multiplication map μ is, by definition, a morphism of algebraic varieties $G \times G \to G$. Thus, for any $x \in G$, there is a morphism of algebraic varieties $\gamma_x : G \to G$ given by $\gamma_x(y) = x^{-1}y$. Therefore, for any $x \in G$, there is a ring homomorphism $\lambda_x\colon k[G] \to k[G]$ given by $(\lambda_x(f))(y) = f(x^{-1}y)$ for all $y \in G$. The ring homomorphism λ_x is called *left translation of functions* by x.

Let A be a k-algebra. A *derivation* on A is a k-linear map $\delta : A \to A$ such that

$$\delta(ab) = a\delta(b) + \delta(a)b, \quad a, b \in A.$$

Set $A = k[G]$, and let $\mathrm{Der}(A)$ be the algebra of derivations of A.

Definition 4.3.28. *The* Lie algebra of G *is*

$$\mathfrak{g} := \{\delta \in \mathrm{Der}(A) \mid \delta\lambda_x = \lambda_x\delta \textit{ for all } x \in G\}.$$

One can verify that $\mathfrak{g}$ is a Lie algebra in the sense of Definition 4.3.22.

Remark 4.3.29. Let G be a linear algebraic group over k, and let $\mathfrak{g}$ be its Lie algebra. In [22, §9], Humphreys identifies $\mathfrak{g}$ with the tangent space of G at the identity.

Example 4.3.30. The Lie algebra of $\mathrm{GL}(n,k)$ is the set of all $n \times n$ matrices over k with the standard Lie bracket.

Example 4.3.31. The Lie algebra of $\mathrm{SL}(n,k)$ is the set of all trace-zero $n \times n$ matrices over k, with the standard Lie bracket.

Theorem 4.3.32. *A connected linear algebraic group over the algebraically closed field k of characteristic 0 is semisimple if and only if its Lie algebra is semisimple.*

Proof: See [22, Theorem 13.5]. ∎

4.3.4 Root decomposition and classification

In this subsection, we classify the connected reductive linear algebraic groups over k in terms of root data. We closely follow Humphreys [22] and Springer [39].

We call a morphism of algebraic groups $\phi : G \to G'$ a *rational representation* of G if $G' \simeq \mathrm{GL}(n,k)$, $n > 0$. A *character* of a linear algebraic group G is a morphism of algebraic groups $\chi : G \to \mathbb{G}_m$. The set of all characters $\Lambda(G)$ of G forms a commutative group with pointwise multiplication. The group $\Lambda(G)$ is called the *character group* of G.

Let G be a linear algebraic group over k, with Lie algebra $\mathfrak{g}$. Let D be a diagonalizable subgroup of a linear algebraic group G, and note that homomorphic images of diagonalizable groups are diagonalizable (see [22, Ch. 16, §16.1]). Let $\rho(D)$ be a representation of D in $\mathrm{GL}(V)$, where V is a finite dimensional vector space over k. Then we can write

$$V = \bigoplus_{\lambda \in \Lambda(D)} V_\lambda; \qquad V_\lambda := \{v \in V \mid x \cdot v = \lambda(x)v \text{ for all } x \in D\}.$$

Those $\lambda \in \Lambda(D)$ such that $V_\lambda \neq (0)$ are called the *weights* of D in V. We call V_λ the *weight space* of V with respect to the weight λ.

There is a natural representation $\mathrm{Ad}(G)$ of G on $\mathfrak{g}$ called the *adjoint representation* of G. By [22, Theorem 10.4], $\mathrm{Ad}(g)(x) = gxg^{-1}$ for all $x \in \mathfrak{g}$ and $g \in G$.

The adjoint representation of G restricts to a representation $\mathrm{Ad}(D)$ of D, which we call the *adjoint representation* of D. The weights of the adjoint representation $\mathrm{Ad}(D) \subseteq \mathrm{GL}(\mathfrak{g})$ are called the *roots of G with respect to D*. We denote the set of roots by $\Sigma(D)$. Let $\mathfrak{g}^D$ be the subspace of $\mathfrak{g}$ corresponding to weight 0 with respect to $\mathrm{Ad}(D)$. There is a *root space decomposition*:

$$\mathfrak{g} = \mathfrak{g}^D \oplus \bigoplus_{\alpha \in \Sigma(D)} \mathfrak{g}_\alpha.$$

Assume that G is connected and reductive, and that $D = T$ is a maximal torus in G. Then $\Sigma(T)$ is a root system in the sense of Definition 4.1.1. If G is connected and semisimple, and $D = T$ is a maximal torus in G, then $\mathfrak{g}^D$ is the Cartan subalgebra of the semisimple Lie algebra $\mathfrak{g}$, and the root space decomposition given above is the Cartan decomposition described in Theorem 4.3.26.

Now suppose G is a connected reductive linear algebraic group over k, and let T be a maximal torus sitting inside G. Denote the character lattice of T by Λ, and denote the set of roots of G with respect to T by Σ. Let $N_G(T)$ be the *normalizer* of T in G, and let $C_G(T)$ be the *centralizer* of T in G. By [22, §26.2 Cor. A (b)], since G is connected and reductive, we have $C_G(T) = T$. The *Weyl group* W of G with respect to T is the quotient $W = N_G(T)/T$. By [22, Lemma 24.1], the group W is finite. Moreover, by [22, Ch. 25, Exercise 8], there is an action of W on the character lattice Λ given by the formula

$$w(\lambda)(t) = \lambda(n^{-1}tn), \quad t \in T,\ \lambda \in \Lambda,\ w = nT \in W,$$

and this action is independent of the representative $n \in N_G(T)$. Under this action, W permutes the roots in Σ.

Let $\Lambda^\vee$ be the lattice dual to Λ, and let $\langle \cdot, \cdot \rangle \colon \Lambda \times \Lambda^\vee \to \mathbb{Z}$ be the duality pairing between the two lattices. The image of Σ in $\Lambda^\vee$ will be denoted $\Sigma^\vee$. Set $\Psi = (\Sigma, \Lambda, \Sigma^\vee, \Lambda^\vee)$.

Proposition 4.3.33. *The quadruple Ψ is a reduced root datum in the sense of Definition 4.2.6.*

Proof: See [39, §7.4]. ∎

Theorem 4.3.34. *Let G and G' be two connected reductive linear algebraic groups over k with isomorphic root data. Then $G \simeq G'$. Moreover, given a reduced root datum Ψ, there exists a connected reductive linear algebraic group G with maximal torus T, such that $\Psi = \Psi(G, T)$.*

Proof: See [39, Theorem 9.6.2 and Theorem 10.1.1]. ∎

Let Λ_r (resp. Λ_w) be the root (resp. weight) lattice of the root system Σ. If G is connected and semisimple, then $\Lambda_r \subseteq \Lambda \subseteq \Lambda_w$. If $\Lambda = \Lambda_r$, we say that $G = G^{\mathrm{ad}}$ is of *adjoint type*. If $\Lambda = \Lambda_w$, we say that $G = G^{\mathrm{sc}}$ is *simply connected*. The *fundamental group* of G is the group $\pi(G) = \Lambda_w/\Lambda$. If $G = G^{\mathrm{ad}}$, then $\pi(G)$ is a fundamental group in the sense of Section 4.2.

Definition 4.3.35. *(See [22, pp. 168]) A linear algebraic group $G \neq \{e\}$ over k is* simple *if it is noncommutative and has no closed connected normal subgroups other than itself and $\{e\}$.*

Lemma 4.3.36. *Let G be a connected semisimple linear algebraic group over k. There is a decomposition $G = G_1 \cdots G_n$ into a (not necessarily direct) product of connected simple linear algebraic groups G_i. Fix a maximal torus T_i of G_i. Then the $\Sigma(G_i, T_i)$ are irreducible root systems, and $\Sigma(G, T)$ is the disjoint union of the $\Sigma(G_i, T_i)$.*

Proof: See [22, Thm. 27.5 and Cor. 27.5]. ∎

Theorem 4.3.37. *Let G and G' be two simple linear algebraic groups over k with isomorphic root systems and isomorphic fundamental groups. Then $G \simeq G'$.*

Proof: See [22, Theorem 32.1]. ∎

A detailed classification of the classical simple linear algebraic groups is given in [39, Ch. 17]. We summarize it below.

- A_n: The groups are in one-to-one correspondence with the positive integers dividing $n+1$. We have

  $$G = \mathrm{SL}(n+1, k)/\mu_l, \quad \text{where } l|(n+1).$$

  In this case,

  $$G^{\mathrm{sc}} = \mathrm{SL}(n+1, k) \quad \text{and} \quad G^{\mathrm{ad}} = \mathrm{PGL}(n+1, k).$$

- B_n: There are exactly two groups:

  $$G^{\mathrm{sc}} = \mathrm{Spin}(2n+1, k) \quad \text{and} \quad G^{\mathrm{ad}} = \mathrm{SO}(2n+1, k).$$

- C_n: There are exactly two groups:

$$G^{\text{sc}} = \text{Sp}(2n, k) \quad \text{and} \quad G^{\text{ad}} = \text{PSp}(2n, k).$$

- D_n, n odd: There are exactly three groups:

$$G^{\text{sc}} = \text{Spin}(2n, k) \quad \text{and} \quad G^{\text{ad}} = \text{PGO}(2n, k) \quad \text{and}$$

$$G = \text{SO}(2n, k) \text{ otherwise.}$$

- D_n, n even: There are exactly five groups:

$$G^{\text{sc}} = \text{Spin}(2n, k) \quad \text{and} \quad G^{\text{ad}} = \text{PGO}(2n, k) \quad \text{and}$$

$$G = \text{SO}(2n, k) \text{ or } G = \text{SSpin}(2n, k)^{+} \text{ or } G = \text{SSpin}(2n, k)^{-} \text{ otherwise.}$$

4.4 Line bundles over flag varieties

The goal of this section is to define the flag variety and construct line bundles over complete flag varieties. We closely follow Borel [3], Humphreys [22], and Springer [39].

Let X be an algebraic variety over k, and suppose there is an action $G \times X \to X$ of a linear algebraic group G over k on X. This action is *transitive* if, for all $x, y \in X$, there is an element $g \in G$ such that $g \cdot x = y$.

Definition 4.4.1. *An algebraic variety X over k on which G acts transitively is a* homogeneous G-variety*.*

Remark 4.4.2. By [39, Thm. 4.3.7], any homogeneous G-variety is smooth.

A closed subgroup P of G is *parabolic* if G/P is a projective variety. Let G be a linear algebraic group over k. We would like to better understand the parabolic subgroups of G. Let H be any closed subgroup of G. By [22, Theorem 11.2], there is a rational representation $\phi : G \to \text{GL}(V)$ and a one-dimensional subspace L of V such that

$$H = \{x \in G \mid \phi(x)L = L\}.$$

In other words, H is the stabilizer of the line L in V. Thus, set-theoretically, one is able to identify the quotient group G/H with the G-orbit of the point $[L]$ in $\mathbb{P}(V)$. Furthermore, one can show that this G-orbit is a quasi-projective variety (see [22, §12]). The following theorem is known as the Borel Fixed Point Theorem:

Theorem 4.4.3. *Let G be a connected solvable algebraic group, and let X be a (nonempty) complete algebraic variety on which G acts. Then G has a fixed point in X.*

Proof: See [22, Theorem 21.2]. ∎

The Borel Fixed Point Theorem is a key ingredient in the proof of the next two results:

Theorem 4.4.4. *Given a Borel subgroup B of G, the quotient G/B has the structure of a projective variety, and the variety G/B is independent of the choice of Borel subgroup B.*

Proof: See [22, Thm. 21.3]. ∎

Using Theorem 4.4.4, we obtain the following (see [22, pp. 135]):

Theorem 4.4.5. *Let H be a closed subgroup of G. The homogeneous G-variety G/H is a projective variety if and only if H contains a Borel subgroup of G.*

Proof: First suppose H contains a Borel subgroup B of G. Then $G/B \to G/H$ is a surjective morphism of algebraic varieties over k. Thus, by Proposition 3.2.3 and Theorem 4.4.4, G/H is projective.

Now suppose G/H is a projective variety, and let B be a Borel subgroup of G. By Proposition 3.2.3, G/H is complete, so the Borel Fixed Point Theorem implies that G/P has a B-fixed point. Thus, there is $g \in G$ such that $bgP = gP$ for all $b \in B$. This implies that $g^{-1}BgP \subseteq P$. So P contains a conjugate of a Borel subgroup of G. Since all Borel subgroups in G are conjugate, this completes the proof. ∎

Definition 4.4.6. *Given any Borel subgroup B of G, the projective variety G/B is called a* complete flag variety*.*

Remark 4.4.7. Let $\mathcal{B}$ be the set of Borel subgroups in G. By the discussion in [22, §23.3], the map $xB \to xBx^{-1}$ defines a bijection $G/B \to \mathcal{B}$. Under the induced variety structure, we call $\mathcal{B}$ the *variety of Borel subgroups* of G.

Let $V = k^n$. The *Grassmannian* $\mathbf{Gr}(r, V)$ is the set of r-dimensional subspaces of V. There is a closed embedding

$$i : \mathbf{Gr}(r, V) \to \mathbb{P}(\Lambda^r V), \quad \mathrm{span}(v_1, \dots, v_r) \mapsto [v_1 \wedge \cdots \wedge v_r],$$

where $\Lambda^r V$ is the r-th exterior power of V, and $[v_1 \wedge \cdots \wedge v_r]$ is the class of the point $v_1 \wedge \cdots \wedge v_r$ in $\Lambda^r V$. Through this embedding, we can view $\mathbf{Gr}(r, V)$ as a projective variety over k. This embedding is called the *Plücker embedding*.

Example 4.4.8. Let V be an n-dimensional vector space over k. A *complete flag* in V is a sequence of subspaces in V,

$$V_1 \subseteq V_2 \subseteq \cdots \subseteq V_n,$$

such that $\dim(V_i) = i$. One can identify the set of complete flags in V with a closed subvariety of a product of Grassmannians. In particular, the set of complete flags in V is a projective variety. If $G = \mathrm{GL}(n,k)$ and B is the Borel subgroup of G, i.e., the subgroup of G consisting of the upper triangular matrices, then the flag variety G/B can be identified with the variety of complete flags in V. See [22, §1.8] for further details.

Let G be a connected semisimple linear algebraic group over k, and suppose T is a maximal torus in G. Let B be a Borel subgroup of G containing T. If λ is a character of T, then λ determines a one-dimensional irreducible representation $V_\lambda = (v_\lambda)$ of T. By Theorem 4.3.17, B/B_u is a torus, and there is sequence of homomorphisms $B \to B/B_u \to T$, where the second homomorphism is an isomorphism. Thus, every character of T lifts to a character λ of B. The group B acts on V_λ by $b \cdot v_\lambda = \lambda(b)^{-1} v_\lambda$ for all $b \in B$.

Theorem 4.4.9. *Let λ be a character of T, and, hence, of B. The set*

$$\mathcal{L}(\lambda) := G \times_B V_\lambda = G \times V_\lambda / ((g,v) \sim (gb, b^{-1} \cdot v))$$

is an algebraic variety, and it is the total space of a line bundle over G/B. The morphism $\pi\colon \mathcal{L}(\lambda) \to G/B$ defining this line bundle sends $(g,v) \mapsto gB$ for all $(g,v) \in \mathcal{L}(\lambda)$.

Proof: See [39, §8.5]. ∎

Remark 4.4.10. There is a natural G-action on $\mathcal{L}(\lambda)$ given by $h \cdot (g,v) = (hg, v)$. The line bundle $\pi : \mathcal{L}(\lambda) \to G/B$ is G-*equivariant*: $\pi(h \cdot (g,v)) = h\pi(g,v)$ for all $g, h \in G$. The line bundle $\mathcal{L}(\lambda)$ is a *homogeneous line bundle.*

Let G and G_1 be connected semisimple linear algebraic groups over k, with root data $\Psi = (\Sigma, \Lambda, \Sigma^\vee, \Lambda^\vee)$ and $\Psi_1 = (\Sigma_1, \Lambda_1, \Sigma_1^\vee, \Lambda_1^\vee)$, and Weyl groups $W(\Psi)$ and $W(\Psi_1)$, respectively. The discussion from here up to Proposition 4.4.15 is taken from [39, §9.6].

Definition 4.4.11. *A* central isogeny *of root data $f : \Psi \to \Psi_1$ is an injective group homomorphism $f : \Lambda \to \Lambda_1$ with finite cokernel such that f induces a bijection $f|_\Sigma\colon \Sigma \to \Sigma_1$, satisfying*

$$f^\vee((f(\alpha))^\vee) = \alpha^\vee, \quad \alpha \in \Sigma.$$

Remark 4.4.12. A central isogeny of root data $f : \Psi \to \Psi_1$ induces an isomorphism of the Weyl groups,

$$W(\Psi) \to W(\Psi_1), \quad s_\alpha \mapsto s_{f(\alpha)}, \quad \alpha \in \Sigma.$$

Remark 4.4.13. Let $\{\alpha_i\}_{i=1}^n$ be a set of simple roots in Σ, and let $\{\lambda_i\}_{i=1}^n$ be a $\mathbb{Z}$-basis of Λ. If $f : \Psi \to \Psi_1$ is a central isogeny, then $\{f(\alpha_i)\}_{i=1}^n$ is a set of simple roots in $f(\Sigma) = \Sigma_1$, and $\{f(\lambda_i)\}_{i=1}^n$ is a $\mathbb{Z}$-basis of $f(\Lambda)$.

Definition 4.4.14. *A* central isogeny $\phi : G_1 \to G$ *is a surjective morphism whose kernel is finite and central in* G_1.

Proposition 4.4.15. *Let $\phi : G_1 \to G$ be an central isogeny, mapping T_1 to T. Then ϕ induces a central isogeny of root data $f : \Psi \to \Psi_1$ such that $f(\lambda) = \lambda \circ \phi|_{T_1}$ for all $\lambda \in \Lambda$.*

Let G^{sc} be the connected semisimple simply-connected linear algebraic group over k with the same Dynkin type as G, and let $\Psi^{\text{sc}} = (\Sigma^{\text{sc}}, \Lambda^{\text{sc}}, (\Sigma^{\text{sc}})^\vee, (\Lambda^{\text{sc}})^\vee)$ be its root datum. By [39, Exercises 10.1.4(1)], there is a central isogeny $\phi : G^{\text{sc}} \to G$. The group G^{sc} is called the *simply-connected cover* of G. If B^{sc} is a Borel subgroup of G^{sc} with maximal unipotent connected subgroup B_u^{sc}, then, by Lemma 4.3.18, $B := \phi(B^{\text{sc}})$ is a Borel subgroup in G with maximal unipotent connected subgroup $B_u := \phi(B_u^{\text{sc}})$. By Theorem 4.3.17, we can view $B^{\text{sc}}/B_u^{\text{sc}}$ and B/B_u as maximal tori in G^{sc} and G, respectively. Set $T^{\text{sc}} := B^{\text{sc}}/B_u^{\text{sc}}$ and $T := \phi(T^{\text{sc}}) = B/B_u$. Let $f : \Lambda \to \Lambda^{\text{sc}}$ be the injective homomorphism on character lattices induced by ϕ. If $p : B \to T$ and $p^{\text{sc}} : B^{\text{sc}} \to T^{\text{sc}}$ are the canonical projections onto the quotients, then the following diagram commutes:

$$\begin{array}{ccc} B^{\text{sc}} & \xrightarrow{p^{\text{sc}}} & T^{\text{sc}} \\ {\scriptstyle \phi|_{B^{\text{sc}}}}\downarrow & & \downarrow{\scriptstyle \phi|_{T^{\text{sc}}}} \\ B & \xrightarrow[p]{} & T \end{array}.$$

Recall that we can lift a character of T^{sc} (resp. T) to a character of B^{sc} (resp. B) by composing the character on the right by p^{sc} (resp. p). Given a character λ of T, we have by Proposition 4.4.15 that $\lambda \circ \phi|_{T^{\text{sc}}} = f(\lambda)$. Thus, $\lambda \circ p \circ \phi|_{B^{\text{sc}}} = \lambda \circ \phi|_{T^{\text{sc}}} \circ p^{\text{sc}} = f(\lambda) \circ p^{\text{sc}}$.

Let $\mathcal{B}$ (resp. $\mathcal{B}^{\text{sc}}$) be the variety of Borel subgroups in G (resp. G^{sc}). Following [3, Prop. 11.20], we show that there is an isomorphism of flag varieties $G^{\text{sc}}/B^{\text{sc}} \simeq G/B$ by showing that the induced map on the variety of Borel subgroups $\phi_{\mathcal{B}^{\text{sc}}} : \mathcal{B}^{\text{sc}} \to \mathcal{B}$ is an isomorphism. Since ϕ is surjective, given $xBx^{-1} \in \mathcal{B}$, there is $y \in G^{\text{sc}}$ such that $\phi(y) = x$. Thus, $\phi_{\mathcal{B}^{\text{sc}}}(yB^{\text{sc}}y^{-1}) = \phi(y)\phi(B^{\text{sc}})\phi(y)^{-1} = xBx^{-1}$. To see that $\phi_{\mathcal{B}^{\text{sc}}}$ is injective, we note that, since the kernel of a central isogeny is central and central elements in G^{sc} lie in B^{sc}, we have

$$\phi^{-1}(B) = B^{\text{sc}} \ker \phi = B^{\text{sc}}.$$

It is well-known that $\phi_{\mathcal{B}^{\mathrm{sc}}}$ and its inverse are morphisms of varieties.

Since $G^{\mathrm{sc}}/B^{\mathrm{sc}} \simeq G/B$ as flag varieties, given a character λ^{sc} of T^{sc}, there is a line bundle $\mathcal{L}(\lambda^{\mathrm{sc}})$ over G/B. If $\lambda^{\mathrm{sc}} \in \Lambda^{\mathrm{sc}}$, $\lambda \in \Lambda$, and $\lambda^{\mathrm{sc}} = f(\lambda)$, then it is well-known that there is an isomorphism of line bundles $\mathcal{L}(\lambda^{\mathrm{sc}}) \simeq \mathcal{L}(\lambda)$ over G/B.

Chapter 5

Oriented cohomology rings of the semisimple linear algebraic groups of ranks 1 and 2

We begin this chapter by defining the formal group algebra $R[\![\Lambda]\!]_F$, which depends on a root datum $\Psi = (\Sigma, \Lambda, \Sigma^\vee, \Lambda^\vee)$ and a one-dimensional commutative formal group law F over a commutative unital ring R. Using the formal group algebra, we define the formal affine Demazure algebra $\mathbf{D}_F$. Suppose $\mathcal{D}_F$ is the subalgebra of the endomorphism algebra of $R[\![\Lambda]\!]_F$ generated by formal Demazure operators and by multiplication by elements in $R[\![\Lambda]\!]_F$. If R satisfies Assumption 5.1.6, then there is an R-algebra isomorphism $\mathbf{D}_F \simeq \mathcal{D}_F$, and there is an R-coalgebra structure on $\mathcal{D}_F$. Thus, there is an R-algebra structure on the dual $\mathcal{D}_F^*$. Given an oriented cohomology theory h^* satisfying Assumption 5.3.3 with formal group law F over R, and a semisimple linear algebraic group over an algebraically closed field k of characteristic 0 with associated root datum Ψ, we construct an algebraic model for the oriented cohomology ring $\mathrm{h}^*(G)$. We conclude this chapter by giving a minimal presentation for $\mathrm{h}^*(G)$ when G is a semisimple linear algebraic group of rank 1 or 2 over an algebraically closed field of characteristic 0. In Section 5.1, we define the formal group algebra and the formal affine Demazure algebra. In Section 5.2, we define the algebra $\mathcal{D}_F$ and discuss the product structure on the dual $\mathcal{D}_F^*$. In Section 5.3, we provide a presentation for the oriented cohomology ring $\mathrm{h}^*(G)$ in terms of formal Demazure operators. In Section 5.4, we compute a minimal presentation for $\mathrm{h}^*(G)$ in terms of generators and relations, where G is a semisimple linear algebraic group of rank 1 or 2.

In this chapter, we fix the following **Notation**: Unless otherwise stated, a *semisimple* algebraic group is a *semisimple and connected* algebraic group. We fix once and for all an algebraically closed field k of characteristic 0.

5.1 Formal affine Demazure algebra

In this section, we define the formal group algebra and the formal affine Demazure algebra. We closely follow [7], [8], and [20].

Let $F = F(u,v) \in R[\![u,v]\!]$ be a one-dimensional commutative formal group law over a commutative, unital ring R. We use the notation

$$u +_F v := F(u,v), \quad m \cdot_F u := \underbrace{u +_F \cdots +_F u}_{m\text{-times}}, \quad \text{and} \quad (-m) \cdot_F u := -_F(m \cdot_F u),$$

where $-_F u$ is the formal inverse of u. Let $\Psi = (\Sigma, \Lambda, \Sigma^\vee, \Lambda^\vee)$ be a semisimple root datum, and let $R[x_\Lambda]$ be the polynomial ring over R with variables indexed by Λ. The augmentation map $\epsilon : R[x_\Lambda] \to R$ sends x_λ to 0 for each $\lambda \in \Lambda$ and fixes R. Let $R[\![x_\Lambda]\!]$ be the $\ker(\epsilon)$-adic completion of the polynomial ring $R[x_\Lambda]$, and let $\mathcal{J}_F$ be the closure of the ideal in $R[\![x_\Lambda]\!]$ generated by x_0 and elements of the form $x_{\lambda_1+\lambda_2} - (x_{\lambda_1} +_F x_{\lambda_2})$ for all $\lambda_1, \lambda_2 \in \Lambda$.

Definition 5.1.1. *The* formal group algebra *is the quotient*

$$R[\![\Lambda]\!]_F := R[\![x_\Lambda]\!]/\mathcal{J}_F.$$

The formal group algebra $R[\![\Lambda]\!]_F$ is a complete Hausdorff ring with respect to the $\mathcal{I}_F$-adic topology, where $\mathcal{I}_F$ is the kernel of the augmentation map $R[\![\Lambda]\!]_F \to R$. By [7, Cor. 2.13], if $\{\lambda_i\}$ is a $\mathbb{Z}$-basis of Λ, then there is an R-algebra isomorphism $R[\![\Lambda]\!]_F \simeq R[\![x_1, \ldots, x_n]\!]$ given by

$$x_{\sum m_i \lambda_i} \mapsto m_1 \cdot_F x_1 +_F \cdots +_F m_n \cdot_F x_n.$$

There is an action of the Weyl group W on $R[\![\Lambda]\!]_F$, where $w \in W$ acts by

$$w(x_\lambda) = x_{w(\lambda)}, \quad \lambda \in \Lambda.$$

Observe that, for each element $\lambda \in \Lambda$, we have $x_{-\lambda} = -_F x_\lambda$.

Example 5.1.2. (a) Let F_A be the *additive* formal group law $F_A(x,y) = x + y$ over R. There is an isomorphism

$$R[\![\Lambda]\!]_{F_A} \to S_R^*(\Lambda) = \prod_{i=0}^{\infty} S_R^i, \quad x_\lambda \mapsto \lambda \in S_R^1(\Lambda) \text{ for all } \lambda \in \Lambda,$$

where S_R^i is the i-th symmetric power of Λ over R.

(b) Let F_M be the *multiplicative periodic* formal group law $F_M(x,y) = x + y - \beta xy$ over R, where $\beta \in R^\times$. Consider the group ring

$$R[\Lambda] := \Big\{ \sum_j r_j e^{\lambda_j} \mid r_j \in R \text{ and } \lambda_j \in \Lambda \Big\}.$$

Let $\mathrm{tr} : R[\Lambda] \to R$ be the trace map, which sends $e^\lambda \mapsto 1$ for all $\lambda \in \Lambda$ and fixes R. We denote the ker(tr)-adic completion of $R[\Lambda]$ by $R[\Lambda]^\wedge$. There is an R-algebra isomorphism

$$R[\![\Lambda]\!]_{F_M} \simeq R[\Lambda]^\wedge,$$

induced by $x_\lambda \mapsto \beta^{-1}(1 - e^{-\lambda})$ and $e^\lambda \mapsto (1 - \beta x_{-\lambda})$.

We say an element $r \in R[\![\Lambda]\!]_F$ is *regular* if it is neither a left nor a right zero divisor in $R[\![\Lambda]\!]_F$.

Definition 5.1.3. *We say that the formal group algebra $R[\![\Lambda]\!]_F$ is* Σ-regular *if x_α is regular in $R[\![\Lambda]\!]_F$ for all $\alpha \in \Sigma$.*

Remark 5.1.4. By [8, Remark 4.5], $R[\![\Lambda]\!]_F$ is Σ-regular if 2 is regular in R or the root datum does not have an irreducible component of type C_n^{sc}, $n \geq 1$.

Let $\mathfrak{t} \in \mathbb{Z}$ be the *torsion index* of the root datum Ψ, as introduced in Demazure [10]. The prime divisors of $\mathfrak{t}$ are the *torsion primes* of the corresponding simply connected root datum, together with the prime divisors of $|\Lambda_w/\Lambda_r|$. We copy the table of prime divisors from [8, §2].

Root system	A_l	B_l, $l \geq 3$	C_l	D_l, $l \geq 3$	G_2	F_4	E_6	E_7	E_8
$\lvert\Lambda_w/\Lambda_r\rvert$	$l+1$	2	2	4	1	1	3	2	1
Torsion primes	$\emptyset$	2	$\emptyset$	2	2	2, 3	2, 3	2, 3	2, 3, 5

Remark 5.1.5. The torsion index has been computed explicitly for all simply connected root data (see [10], [40], and [41]).

For the remainder of this section, we work under the following assumption.

Assumption 5.1.6. We assume $R[\![\Lambda]\!]_F$ is Σ-regular, and that $\mathfrak{t}$ is regular in R.

Remark 5.1.7. Observe that, since $|\Lambda_w/\Lambda_w| = 1$, if $\mathfrak{t}$ is regular in R, then the torsion index of the corresponding simply connected root datum is regular in R.

Lemma 5.1.8. *For each $u \in R[\![\Lambda]\!]_F$ and root α, the element $u - s_\alpha(u)$ is uniquely divisible by x_α.*

Proof: This follows from [8, Lemma 4.1]. ∎

Definition 5.1.9. *For each root $\alpha \in \Sigma$, define the formal Demazure operator*

$$\Delta_\alpha : R[\![\Lambda]\!]_F \to R[\![\Lambda]\!]_F, \qquad \Delta_\alpha(u) = \frac{u - s_\alpha(u)}{x_\alpha}.$$

Remark 5.1.10. The formal Demazure operator satisfies a *twisted Leibnitz rule*. For all $u, v \in R[\![\Lambda]\!]_F$, we have

$$\Delta_\alpha(uv) = \Delta_\alpha(u)v + s_\alpha(u)\Delta_\alpha(v), \quad \alpha \in \Sigma.$$

Note also that, for all $\alpha \in \Sigma$, we have $\Delta_\alpha(\mathcal{I}_F^i) \subseteq \mathcal{I}_F^{i-1}$ for all $i > 0$.

Definition 5.1.11. *Let $\mathcal{Q}^F$ be the localization of $R[\![\Lambda]\!]_F$ at the multiplicative set generated by the regular elements $\{x_\alpha \mid \alpha \in \Sigma\}$. The action of W on $R[\![\Lambda]\!]_F$ induces an action by automorphisms on $\mathcal{Q}^F$. The* twisted formal group algebra *$\mathcal{Q}_W^F$ is the tensor product $\mathcal{Q}^F \otimes_R R[W]$ as an R-module, with multiplication*

$$(q\delta_w)(q'\delta_{w'}) = qw(q')\delta_{ww'}, \quad q, q' \in \mathcal{Q}^F, w, w' \in W.$$

Here, δ_w is the element in the group ring $R[W]$ corresponding to $w \in W$.

Definition 5.1.12. *For each root $\alpha \in \Sigma$, we define the* formal Demazure element*,*

$$X_\alpha = \tfrac{1}{x_\alpha}(1 - \delta_{s_\alpha}) \in \mathcal{Q}_W^F.$$

Definition 5.1.13. *The* formal affine Demazure algebra $\mathbf{D}_F$ *is the R-subalgebra of $\mathcal{Q}_W^F$ generated by $R[\![\Lambda]\!]_F$ and by the formal Demazure elements X_α, $\alpha \in \Sigma$.*

For $i > 0$, define $[i] := \{1, \ldots, i\}$. Let $\Delta = \{\alpha_1, \ldots, \alpha_n\}$ be a simple system for Σ, with corresponding simple reflections $\{s_1, \ldots, s_n\}$. Suppose $m_{i,j}$ is the order of $s_i s_j$ in W. For $i \in [n]$, we set $\Delta_i = \Delta_{\alpha_i}$ and $X_i = X_{\alpha_i}$. If $I = (i_1, \ldots, i_t)$ is a sequence in $[n]$, we define its length $l(I) = t$, and we set

$$s_I = s_{i_1} \cdots s_{i_t} \quad \text{and} \quad \Delta_I = \Delta_{i_1} \circ \cdots \circ \Delta_{i_t} \quad \text{and} \quad X_I = X_{i_1} \cdots X_{i_t}.$$

We call I a *sequence of simple roots*. We say that I is *reduced* if $w(I) := s_{i_1} \cdots s_{i_t}$ is reduced in W. For any reduced decomposition $w = s_{i_1} \cdots s_{i_t}$ of $w \in W$, we call $I_w = (i_1, \ldots, i_t)$ a reduced sequence of w.

For each $w \in W$, fix a reduced sequence I_w. Set $w_0^{i,j} := \underbrace{s_i s_j s_i \cdots}_{m_{i,j}\text{-times}}$.

Proposition 5.1.14. *The R-algebra $\mathbf{D}_F$ is free as a left $R[\![\Lambda]\!]_F$-module, with basis $\{X_{I_w}\}_{w \in W}$.*

Proof: See [8, Prop. 7.7]. ∎

The following remark is taken from [20, Def. 4.2].

Remark 5.1.15. For $\alpha \in \Sigma$, set $\kappa_\alpha := \frac{1}{x_\alpha} + \frac{1}{x_{-\alpha}} \in \mathcal{Q}^F$. Since

$$0 = x_\alpha +_F x_{-\alpha} = x_\alpha + x_{-\alpha} + \sum_{i,j\geq 1} a_{i,j} x_\alpha^i x_{-\alpha}^j,$$

we see that

$$\kappa_\alpha = -\sum_{i,j\geq 1} a_{i,j} x_\alpha^{i-1} x_{-\alpha}^{j-1} \in R[\![\Lambda]\!]_F.$$

Observe that $\epsilon(\kappa_\alpha) = -a_{11}$.

Theorem 5.1.16. *The elements $q \in R[\![\Lambda]\!]_F$ and the formal Demazure elements $X_i = X_{\alpha_i}$, where $\alpha_i \in \Delta$, satisfy the following relations:*

(1) $X_i q = \Delta_i(q) + s_i(q) X_i$;

(2) $X_i^2 = \kappa_i X_i$, *where* $\kappa_i := \frac{1}{x_{\alpha_i}} + \frac{1}{x_{-\alpha_i}} \in R[\![\Lambda]\!]_F$;

(3) $\underbrace{X_i X_j X_i \cdots}_{m_{i,j}\text{-times}} - \underbrace{X_j X_i X_j \cdots}_{m_{i,j}\text{-times}} = \sum_{w < w_0^{i,j}} \eta_w^{i,j} X_{I_w}$, $\eta_w^{i,j} \in R[\![\Lambda]\!]_F$.

Note that the ordering $<$ is with respect to the Bruhat order on W. These relations, together with the ring law in $R[\![\Lambda]\!]_F$ and the fact that the X_i are R-linear form a complete set of relations in $\mathbf{D}_F$.

Proof: See [8, Theorem 7.9]. ∎

Remark 5.1.17. The $\eta_w^{i,j}$ of Theorem 5.1.16 can be computed explicitly from the coefficients computed in [20, Thm. 6.8] using the formulas (1) and (2) of Theorem 5.1.16.

5.2 Formal Demazure operators

In this section, we assume $R[\![\Lambda]\!]_F$ is a formal group algebra satisfying Assumption 5.1.6. In this section, we discuss the subalgebra $\mathcal{D}_F$ of the endomorphism algebra of $R[\![\Lambda]\!]_F$ generated by formal Demazure operators and by multiplication by elements in $R[\![\Lambda]\!]_F$. The dual $\mathcal{D}_F^*$ of $\mathcal{D}_F$ will be used in the upcoming sections to study the oriented cohomology rings of the semisimple linear algebraic groups over k.

Definition 5.2.1. *Let $\mathcal{D}_F$ be the subalgebra of the algebra of R-linear endomorphisms of $R[\![\Lambda]\!]_F$ generated by the elements Δ_α, $\alpha \in \Sigma$, and by multiplication by elements in $R[\![\Lambda]\!]_F$.*

Theorem 5.2.2. *The $R[\![\Lambda]\!]_F$-linear map $\phi : \mathbf{D}_F \to \mathcal{D}_F$ sending $X_i \mapsto \Delta_i$ is an isomorphism of R-algebras.*

Proof: See [8, Theorem 7.10]. ∎

Fix a reduced sequence I_w for each $w \in W$.

Theorem 5.2.3. *The algebra $\mathcal{D}_F$ is free as a left $R[\![\Lambda]\!]_F$-module with basis $\{\Delta_{I_w}\}_{w\in W}$.*

Proof: This follows from Proposition 5.1.14 and Theorem 5.2.2. ∎

Let q^* be the operator in $\mathcal{D}_F$ corresponding to multiplication by $q \in R[\![\Lambda]\!]_F$.

Proposition 5.2.4. *The elements $q \in R[\![\Lambda]\!]_F$ and the formal Demazure operators $\Delta_i = \Delta_{\alpha_i}$, where $\alpha_i \in \Delta$, satisfy the following relations:*

(1) $\Delta_i \circ q^* = (\Delta_i(q))^* + (s_i(q))^* \circ \Delta_i$;

(2) $\Delta_i^2 = \kappa_i^* \circ \Delta_i$, *where* $\kappa_i := \frac{1}{x_{\alpha_i}} + \frac{1}{x_{-\alpha_i}} \in R[\![\Lambda]\!]_F$;

(3) $\underbrace{\Delta_i \circ \Delta_j \circ \Delta_i \cdots}_{m_{i,j}\text{-times}} - \underbrace{\Delta_j \circ \Delta_i \circ \Delta_j \cdots}_{m_{i,j}\text{-times}} = \sum_{w<w_0^{i,j}} (\eta_w^{i,j})^* \circ \Delta_{I_w}$, $\eta_w^{i,j} \in R[\![\Lambda]\!]_F$.

Note that the ordering $<$ is with respect to the Bruhat order on W. These relations, together with the ring law in $R[\![\Lambda]\!]_F$ and the fact that the Δ_i are R-linear form a complete set of relations in $\mathcal{D}_F$.

Proof: This follows from Theorem 5.1.16 and Theorem 5.2.2. ∎

Definition 5.2.5. *We define R-linear operators $B_i^{(j)} \colon R[\![\Lambda]\!]_F \to R[\![\Lambda]\!]_F$, where $j \in \{0, 1, -1\}$ and $i \in [n]$, by*

$$B_i^{(-1)} := \Delta_i, \quad B_i^{(0)} := s_i, \quad B_i^{(1)} := \text{multiplication by } (-x_i) := -x_{\alpha_i}.$$

Let $I = (i_1, \ldots, i_t)$ be a sequence of simple roots, and let E be a subset of $[t]$. We denote by I_E the subsequence of I consisting of all i_j's with $j \in E$.

Proposition 5.2.6. *There is a cocommutative coproduct on $\mathcal{D}_F$, denoted $\triangle$, with counit η. Given any sequence of simple roots $I = (i_1, \ldots, i_t)$, the coproduct satisfies*

$$\triangle(\Delta_I) = \sum_{E_1, E_2 \subseteq [t]} p^I_{E_1,E_2} \Delta_{I|_{E_1}} \otimes \Delta_{I|_{E_2}},$$

where $p^I_{E_1,E_2} := B_1 \circ \cdots \circ B_t(1)$, *and the operator* $B_j : R[\![\Lambda]\!]_F \to R[\![\Lambda]\!]_F$ *is defined by*

$$B_j = \begin{cases} B^{(1)}_{i_j} \circ B^{(0)}_{i_j}, & \text{if } j \in E_1 \cap E_2; \\ B^{(-1)}_{i_j}, & \text{if } j \notin E_1 \cup E_2; \\ B^{(0)}_{i_j}, & \text{otherwise.} \end{cases}$$

Proof: This follows from [8, Prop. 9.5 and Thm. 10.4]. ∎

Let $\mathcal{D}^*_F = \mathrm{Hom}(\mathcal{D}_F, R[\![\Lambda]\!]_F)$ be the $R[\![\Lambda]\!]_F$-linear dual of $\mathcal{D}_F$. The cocommutative coalgebra structure on $\mathcal{D}_F$ induces a commutative algebra structure on $\mathcal{D}^*_F$. Let $B^* = \{\Delta^*_{I_w}\}_{w\in W}$ be the basis of $\mathcal{D}^*_F$ dual to $B = \{\Delta_{I_w}\}_{w\in W}$. By Theorem 5.2.3 and Proposition 5.2.6, given any $\Delta_{I_v} \in B$, we can write

$$\triangle(\Delta_{I_v}) = \sum_{w,w'\in W} q^v_{w,w'} \Delta_{I_w} \otimes \Delta_{I_{w'}} \ , \tag{5.2.1}$$

where the $q^v_{w,w}$ are $\mathbb{Z}$-linear combinations of products of the $p^I_{E_1,E_2}$ of Proposition 5.2.6, and the structure coefficients of Proposition 5.2.4. Thus, products of elements in the dual basis satisfy

$$\Delta^*_{I_w} \cdot \Delta^*_{I_{w'}} = \sum_{v\in W} q^v_{w,w'} \Delta^*_{I_v}.$$

Example 5.2.7. Suppose Σ is a root system of rank 1. Then $\Sigma = \{\alpha_1, -\alpha_1\}$, and $W = \{e, s_1\}$. We compute

$$\triangle(\Delta_1) = \mathbf{1} \otimes \Delta_1 + \Delta_1 \otimes \mathbf{1} - x_{\alpha_1} \Delta_1 \otimes \Delta_1 \quad \text{and} \quad \triangle(\mathbf{1}) = \mathbf{1} \otimes \mathbf{1}.$$

Thus,

$$\Delta^*_1 \cdot \Delta^*_1 = -x_{\alpha_i} \Delta^*_1 \quad \text{and} \quad \mathbf{1}^* \cdot \Delta^*_1 = \Delta^*_1 \cdot \mathbf{1}^* = \Delta^*_1.$$

5.3 Structure of oriented cohomology rings of semisimple linear algebraic groups

In this section, we fix a semisimple linear algebraic group G, and a maximal torus T sitting inside a Borel subgroup B of G. In this section, we provide a presentation for the oriented cohomology ring $\mathrm{h}^*(G)$ in terms of formal Demazure operators, where h^* is any suitable oriented cohomology theory. Before we proceed, we will briefly explain [7, Assumption 13.2]:

Remark 5.3.1. Let Σ be the root system of T, with Weyl group W, and fix a set of simple roots $\{\alpha_1, \dots, \alpha_n\}$ of Σ. Given a sequence of simple roots $I = (\alpha_{i_1}, \dots, \alpha_{i_l})$, there is a variety X_I called the *Bott-Samelson* variety corresponding to I. The quotient X_I/B exists, and there is a morphism $q_I : X_I/B \to G/B$. If $\zeta_I := (q_I)_*(1) \in \mathrm{h}^*(G/B)$ is the push-forward of the fundamental class of the Bott-Samelson variety corresponding to I, then [7, Assumption 13.2] says that, given a reduced sequence I_w for each $w \in W$, the set $\{\zeta_{I_w}\}_{w \in W}$ is an R-basis of $\mathrm{h}^*(G/B)$, where $R = \mathrm{h}^*(\mathrm{pt})$.

The following definition is taken from [7, Def. 8.7].

Definition 5.3.2. *We say that an oriented cohomology theory* h^* *is* weakly birationally invariant *if, for any proper birational morphism of varieties* $f : Y \to X$*, the pushforward of the fundamental class* $f_*(1_Y)$ *is invertible.*

In order for the main results in this section to hold, the theory h^* must satisfy Assumption 5.3.3 below:

Assumption 5.3.3. h^* is a weakly birationally invariant theory satisfying [7, Assumption 13.2].

Since k has characteristic 0, we have by [7, Ex. 8.8 and Lem. 13.3] that any oriented cohomology theory (including algebraic cobordism Ω^*) satisfies Assumption 5.3.3. For the rest of this thesis, we fix the following notation and conventions.

- $\Psi = (\Sigma, \Lambda, \Sigma^\vee, \Lambda^\vee)$ is a reduced root datum, with Weyl group W and torsion index $\mathfrak{t}$.
- $\Delta = \{\alpha_i\}_{i=1}^n$ is a simple system for Σ, and $\{\lambda_i\}_{i=1}^n$ is a basis for Λ.
- G is a semisimple linear algebraic group over k, with root datum Ψ.
- B is a Borel subgroup of G, so that G/B is a complete flag variety.
- $\phi : G^{\mathrm{sc}} \to G$ is the simply connected cover of G.
- $\Psi^{\mathrm{sc}} = (\Sigma^{\mathrm{sc}}, \Lambda^{\mathrm{sc}}, (\Sigma^{\mathrm{sc}})^\vee, (\Lambda^{\mathrm{sc}})^\vee)$ is the root datum of G^{sc}, with Weyl group W^{sc} and torsion index $\mathfrak{t}^{\mathrm{sc}}$.
- $\sigma : \Lambda \to \Lambda^{\mathrm{sc}}$ is the group homomorphism on character lattices induced by ϕ.
- B^{sc} is the Borel subgroup in G^{sc} corresponding to B, and T^{sc} is the maximal torus in G^{sc} corresponding to T.
- h^* is any oriented cohomology theory, such that $\mathfrak{t}$ is regular in $\mathrm{h}^*(\mathrm{pt})$.
- $F(x, y) = x + y + \sum_{i,j \geq 1} a_{i,j} x^i y^j \in R[\![x, y]\!]$ is the formal group law over $R = \mathrm{h}^*(\mathrm{pt})$ associated to h^*.

- The formal group algebras $R[\![\Lambda]\!]_F$ and $R[\![\Lambda^{\mathrm{sc}}]\!]_F$ satisfy Assumption 5.1.6.
- $\mathcal{D}_F$ (resp. $\mathcal{D}_F^{\mathrm{sc}}$) is the algebra defined in Definition 5.2.1 corresponding to $R[\![\Lambda]\!]_F$ (resp. $R[\![\Lambda^{\mathrm{sc}}]\!]_F$).
- For each $w \in W$, $I_w = (\alpha_{i_1}, \ldots, \alpha_{i_k})$ is a reduced sequence of w.
- $\sigma(I_w) := (\sigma(\alpha_{i_1}), \ldots, \sigma(\alpha_{i_k}))$ is the reduced sequence of the reflection $\sigma(w) := s_{\sigma(\alpha_{i_1})} \cdots s_{\sigma(\alpha_{i_k})} \in W^{\mathrm{sc}}$ corresponding to I_w.

The homomorphism σ induces an injection $\sigma_* : R[\![\Lambda]\!]_F \to R[\![\Lambda^{\mathrm{sc}}]\!]_F$ sending $x_\lambda \mapsto x_{\sigma(\lambda)}$.

Lemma 5.3.4. *Let $s \in R[\![\Lambda]\!]_F$. Then*

$$s_{\sigma(\alpha_i)}(\sigma_*(s)) = \sigma_*(s_{\alpha_i}(s)).$$

Proof: If $s = x_\lambda$ for some $\lambda \in \Lambda$, then

$$s_{\sigma(\alpha_i)}(\sigma_*(x_\lambda)) = x_{\sigma(\lambda)-(\sigma(\alpha_i))^\vee(\sigma(\lambda))\sigma(\alpha_i)} = x_{\sigma(\lambda)-\alpha_i^\vee(\lambda)\sigma(\alpha_i)} = \sigma_*(s_{\alpha_i}(x_\lambda)).$$

By induction on the degree of monomials, for $u, v \in R[\![\Lambda]\!]_F$, we have

$$s_{\sigma(\alpha_i)}(\sigma_*(uv)) = (s_{\sigma(\alpha_i)}(\sigma_*(u)))(s_{\sigma(\alpha_i)}(\sigma_*(v))) = \sigma_*(s_{\alpha_i}(uv)).$$

By density of $R[\![\Lambda]\!]_F$, for all $s \in R[\![\Lambda]\!]_F$, we have $s_{\sigma(\alpha_i)}(\sigma_*(s)) = \sigma_*(s_{\alpha_i}(s))$. ∎

Recall the augmentation map $\epsilon : R[\![\Lambda]\!]_F \to R$ sends $x_\lambda \mapsto 0$ for all $\lambda \in \Lambda$.

Lemma 5.3.5. *Let $s \in R[\![\Lambda]\!]_F$. Then*

$$\epsilon\Delta_{I_w}(s) = \epsilon\Delta_{\sigma(I_w)}(\sigma_*(s)).$$

Proof: By Lemma 5.3.4, we have

$$\Delta_{\sigma(\alpha_i)}(\sigma_*(s)) = \frac{\sigma_*(s) - s_{\sigma(\alpha_i)}(\sigma_*(s))}{x_{\sigma(\alpha_i)}} = \sigma_*\left(\frac{s - s_{\alpha_i}(s)}{x_{\alpha_i}}\right) = \sigma_*\left(\Delta_{\alpha_i}(s)\right).$$

The result follows from the fact that σ_* fixes R. ∎

The discussion from here up to Theorem 5.3.6 closely follows [8, §10]. The augmentation map induces a map of R-modules $\epsilon_* : \mathcal{D}_F \to \mathrm{Hom}_R(R[\![\Lambda]\!]_F, R)$ given by $f \mapsto \epsilon \circ f$. Let $\epsilon\mathcal{D}_F$ be the image of $\mathcal{D}_F$ under the induced map. We can view

$\epsilon\mathcal{D}_F$ as an $R[\![\Lambda]\!]_F$-module, where the $R[\![\Lambda]\!]_F$-action is given by $s \cdot f := \epsilon(s)f$ for all $s \in R[\![\Lambda]\!]_F$ and $f \in \epsilon\mathcal{D}_F$. There is an $R[\![\Lambda]\!]_F$-linear coproduct

$$\triangle^\epsilon : \epsilon\mathcal{D}_F \to \epsilon\mathcal{D}_F \otimes_{R[\![\Lambda]\!]_F} \epsilon\mathcal{D}_F,$$

such that

$$\triangle^\epsilon(f)(u \otimes v) = f(uv), \quad u, v \in R[\![\Lambda]\!]_F.$$

The map $\epsilon_* : \mathcal{D}_F \to \epsilon\mathcal{D}_F$ is a morphism of $R[\![\Lambda]\!]_F$-coalgebras. Since Assumption 5.1.6 holds, Theorem 5.3.6 goes through:

Theorem 5.3.6. *The augmentation $\epsilon\mathcal{D}_F$ (resp. $\epsilon\mathcal{D}_F^{sc}$) is free as a left R-module with basis $\{\epsilon\Delta_{I_w}\}_{w\in W}$ (resp. $\{\epsilon\Delta_{\sigma(I_w)}\}_{w\in W}$).*

Proof: See [7, Thm. 5.4]. ∎

Since ϵ_* is a morphism of $R[\![\Lambda]\!]_F$-coalgebras, we can compute the coproduct $\triangle^\epsilon$ in the basis $\{\epsilon\Delta_{I_w}\}_{w\in W}$ using Equation 5.2.1.

Remark 5.3.7. By Lemma 5.3.4 and Lemma 5.3.5, in the notation of Proposition 5.2.6, we have that

$$\epsilon(p^I_{E_1,E_2}) = \epsilon(p^{\sigma(I)}_{\sigma(E_1),\sigma(E_2)}),$$

where $p^{\sigma(I)}_{\sigma(E_1),\sigma(E_2)}$ are the coproduct coefficients in $\mathcal{D}_F^{\mathrm{sc}}$. Thus, by Theorem 5.3.6, σ induces an isomorphism of R-coalgebras,

$$\epsilon\mathcal{D}_F \to \epsilon\mathcal{D}_F^{\mathrm{sc}}, \quad \epsilon\Delta_{\alpha_i} \mapsto \epsilon\Delta_{\sigma(\alpha_i)}.$$

Therefore,

$$\epsilon\mathcal{D}_F^* \to (\epsilon\mathcal{D}_F^{\mathrm{sc}})^*, \quad \epsilon\Delta^*_{\alpha_i} \mapsto (\epsilon\Delta^{\mathrm{sc}}_{\sigma(\alpha_i)})^*$$

is an isomorphism of R-algebras.

We recall three different maps of R-algebras:

- $\xi : \mathcal{D}_F^* \to (\epsilon\mathcal{D}_F)^*$ (or $\xi^{\mathrm{sc}} : (\mathcal{D}_F^{\mathrm{sc}})^* \to (\epsilon\mathcal{D}_F^{\mathrm{sc}})^*$) is the *surjective* map of R-algebras, defined in [8, Lemma 11.2], which sends f to the map $\epsilon d \mapsto \epsilon f(d)$.
- $c_{R[\![\Lambda]\!]_F} : R[\![\Lambda]\!]_F \to \mathcal{D}_F^*$ (resp. $c_{R[\![\Lambda^{\mathrm{sc}}]\!]_F} : R[\![\Lambda^{\mathrm{sc}}]\!]_F \to (\mathcal{D}_F^{\mathrm{sc}})^*$) is the R-algebra map, defined in [8, pp. 24], which sends s to $\mathrm{ev}_s = \sum_{w\in W} \Delta_{I_w}(s)\Delta^*_{I_w}$ (resp. $\mathrm{ev}_s = \sum_{w\in W} \Delta_{\sigma(I_w)}(s)\Delta^*_{\sigma(I_w)}$).
- $c_R : R[\![\Lambda]\!]_F \to (\epsilon\mathcal{D}_F)^*$ (resp. $c_R^{\mathrm{sc}} : R[\![\Lambda^{\mathrm{sc}}]\!]_F \to (\epsilon\mathcal{D}_F^{\mathrm{sc}})^*$) is the R-algebra map, defined in [7, Theorem 7.3], which sends s to $\sum_{w\in W} \epsilon\Delta_{I_w}(s)(\epsilon\Delta_{I_w})^*$ (resp. $\sum_{w\in W} \epsilon\Delta_{\sigma(I_w)}(s)(\epsilon\Delta_{\sigma(I_w)})^*$).

By [7, Section 6 and Thm. 13.12], for all $\lambda \in \Lambda^{\mathrm{sc}}$, there is a ring homomorphism,

$$\mathfrak{c}_{G^{\mathrm{sc}}/B^{\mathrm{sc}}} : R[\![\Lambda^{\mathrm{sc}}]\!]_F \to \mathrm{h}^*(G^{\mathrm{sc}}/B^{\mathrm{sc}}) \simeq \mathrm{h}^*(G/B), \quad x_\lambda \mapsto \mathrm{c}_1^{\mathrm{h}^*}(\mathcal{L}(\lambda)),$$

where $\mathrm{c}_1^{\mathrm{h}^*}(\mathcal{L}(\lambda))$ is the first Chern class of the line bundle $\mathcal{L}(\lambda)$ with respect to h^*. This gives a ring homomorphism called the *characteristic map*, such that for all $\lambda \in \Lambda$:

$$\mathfrak{c}_{G/B} : R[\![\Lambda]\!]_F \to R[\![\Lambda^{\mathrm{sc}}]\!]_F \to \mathrm{h}^*(G/B), \quad x_\lambda \mapsto \mathrm{c}_1^{\mathrm{h}^*}(\mathcal{L}(\sigma(\lambda))).$$

By definition, $\mathfrak{c}_{G/B} = \mathfrak{c}_{G^{\mathrm{sc}}/B^{\mathrm{sc}}} \circ \sigma^*$.

Theorem 5.3.8. *There is an R-algebra isomorphism $\theta : (\epsilon\mathcal{D}_F^{sc})^* \to \mathrm{h}^*(G/B)$ such that $\mathfrak{c}_{G/B} = \theta \circ c_R^{sc} \circ \sigma_*$.*

Proof: By [7, Theorem 13.12], there exists an R-algebra isomorphism $\theta : (\epsilon\mathcal{D}_F^{\mathrm{sc}})^* \to \mathrm{h}^*(G^{\mathrm{sc}}/B^{\mathrm{sc}})$ such that $\mathfrak{c}_{G^{\mathrm{sc}}/B^{\mathrm{sc}}} = \theta \circ c_R^{\mathrm{sc}}$. The result follows by composing on the right by σ^*. ∎

Theorem 5.3.9. *Given a $\mathbb{Z}$-basis $\{\lambda_i\}_{i=1}^n$ for Λ, there is an R-algebra isomorphism*

$$\begin{aligned}\mathrm{h}^*(G) &\simeq \mathrm{h}^*(G/B)/(c_1^{\mathrm{h}^*}(\mathcal{L}(\lambda_1)), \dots, c_1^{\mathrm{h}^*}(\mathcal{L}(\lambda_n))) \\ &\simeq \mathrm{h}^*(G/B)/(c_1^{\mathrm{h}^*}(\mathcal{L}(\sigma(\lambda_1))), \dots, c_1^{\mathrm{h}^*}(\mathcal{L}(\sigma(\lambda_n)))).\end{aligned}$$

Proof: This follows from [17, Theorem 5.1], and from the fact that $\mathcal{L}(\lambda_i) \simeq \mathcal{L}(\sigma(\lambda_i))$ as line bundles over G/B. ∎

Let $\mathcal{I}_F$ (resp. $\mathcal{I}_F^{\mathrm{sc}}$) be the kernel of the augmentation map $\epsilon : R[\![\Lambda]\!]_F \to R$ (resp. $\epsilon : R[\![\Lambda^{\mathrm{sc}}]\!]_F \to R$). Denote by $\mathcal{I}_F\mathcal{D}_F^*$ (resp. $\mathcal{I}_F^{\mathrm{sc}}(\mathcal{D}_F^{\mathrm{sc}})^*$) the ideal in $\mathcal{D}_F^*$ (resp. $(\mathcal{D}_F^{\mathrm{sc}})^*$) generated by multiplication by elements in $\mathcal{I}_F$ (resp. $\mathcal{I}_F^{\mathrm{sc}}$). Let $\mathcal{C}_{R[\![\Lambda]\!]_F}$ be the ideal in $\mathcal{D}_F^*$ generated by the image of the restriction $c_{R[\![\Lambda]\!]_F}|_{\mathcal{I}_F}$ of $c_{R[\![\Lambda]\!]_F}$ to $\mathcal{I}_F$. The following theorem is the main result of this section. When h^* is the Chow theory, this theorem is well-known (see, for example, [28, pp. 19]).

Theorem 5.3.10. *There is an isomorphism of R-algebras*

$$\mathcal{D}_F^*/(\mathcal{C}_{R[\![\Lambda]\!]_F} + \mathcal{I}_F\mathcal{D}_F^*) \simeq \mathrm{h}^*(G).$$

Proof: Using Theorem 5.2.3 and Theorem 5.3.6, we can verify that the kernel of the surjective map $\xi \colon \mathcal{D}_F^* \to (\epsilon\mathcal{D}_F)^*$ is $\mathcal{I}_F\mathcal{D}_F^*$. Suppose

$$\xi\left(\sum_{w\in W} a_w \Delta_{I_w}^*\right) = 0, \quad a_w \in R[\![\Lambda]\!]_F.$$

Then, for all $v \in W$, we have

$$0 = \left(\xi\left(\sum_{w\in W} a_w \Delta^*_{I_w}\right)\right)(\epsilon\Delta_{I_v}) = \epsilon\left(\left(\sum_{w\in W} a_w \Delta^*_{I_w}\right)(\Delta_{I_v})\right) = \epsilon(a_v).$$

Since $\ker(\epsilon) = \mathcal{I}_F$, we have $\ker(\xi) = \mathcal{I}_F\mathcal{D}^*_F$. Hence, the induced map $\bar{\xi} : \mathcal{D}^*_F/\mathcal{I}_F\mathcal{D}^*_F \to (\epsilon\mathcal{D}_F)^*$ is an R-algebra isomorphism.

Define $\bar{c}_{R[\![\Lambda]\!]_F} = \pi \circ c_{R[\![\Lambda]\!]_F} : R[\![\Lambda]\!]_F \to \mathcal{D}^*_F \to \mathcal{D}^*_F/\mathcal{I}_F\mathcal{D}^*_F$, where $\pi : \mathcal{D}^*_F \to \mathcal{D}^*_F/\mathcal{I}_F\mathcal{D}^*_F$ is the canonical projection. Given $s \in R[\![\Lambda]\!]_F$, we see that

$$\begin{aligned}(c_R(s))(\epsilon\Delta_{I_v}) &= \left(\sum_{w\in W} \epsilon\Delta_{I_w}(s)(\epsilon\Delta_{I_w})^*\right)(\epsilon\Delta_{I_v}) \\ &= \epsilon\Delta_{I_v}(s) = \bar{\xi}\left(\sum_{w\in W} \Delta_{I_w}(s)\Delta^*_{I_w}\right)(\epsilon\Delta_{I_v}) = ((\bar{\xi}\circ \bar{c}_{R[\![\Lambda]\!]_F})(s))(\epsilon\Delta_{I_v}).\end{aligned}$$

Therefore, $c_R = \bar{\xi}\circ\bar{c}_{R[\![\Lambda]\!]_F}$. By analogous reasoning, we have $c^{\mathrm{sc}}_R = \bar{\xi}^{\mathrm{sc}} \circ \bar{c}_{R[\![\Lambda^{\mathrm{sc}}]\!]_F}$

The isomorphism

$$(\epsilon\mathcal{D}_F)^* \to (\epsilon\mathcal{D}^{\mathrm{sc}}_F)^*$$

of Remark 5.3.7 induces an isomorphism

$$\sigma^* : \mathcal{D}^*_F/\mathcal{I}_F\mathcal{D}^*_F \to (\mathcal{D}^{\mathrm{sc}}_F)^*/\mathcal{I}^{\mathrm{sc}}_F(\mathcal{D}^{\mathrm{sc}}_F)^*, \quad \Delta^*_{\alpha_i} + \mathcal{I}_F\mathcal{D}^*_F \mapsto \Delta^*_{\sigma(\alpha_i)} + \mathcal{I}^{\mathrm{sc}}_F(\mathcal{D}^{\mathrm{sc}}_F)^*.$$

By Lemma 5.3.5, under this isomorphism, $\mathrm{im}(\bar{c}_{R[\![\Lambda^{\mathrm{sc}}]\!]_F}\circ\sigma_*)$ in $(\mathcal{D}^{\mathrm{sc}}_F)^*/\mathcal{I}^{\mathrm{sc}}_F(\mathcal{D}^{\mathrm{sc}}_F)^*$ corresponds to $\mathrm{im}(\bar{c}_{R[\![\Lambda]\!]_F})$ in $\mathcal{D}^*_F/\mathcal{I}_F\mathcal{D}^*_F$.

By Theorem 5.3.8, there is an R-algebra isomorphism $\theta : (\epsilon\mathcal{D}^{\mathrm{sc}}_F)^* \to \mathrm{h}^*(G/B)$ such that $\mathfrak{c}_{G/B} = \theta\circ c^{\mathrm{sc}}_R\circ\sigma_*$. In particular, $(\mathcal{D}^{\mathrm{sc}}_F)^*/\mathcal{I}^{\mathrm{sc}}_F(\mathcal{D}^{\mathrm{sc}}_F)^* \simeq \mathrm{h}^*(G/B)$, and $\mathfrak{c}_{G/B} = \theta\circ\bar{\xi}^{\mathrm{sc}}\circ\bar{c}_{R[\![\Lambda^{\mathrm{sc}}]\!]_F}\circ\sigma_*$.

By Theorem 5.3.9, there is an R-algebra isomorphism

$$\mathrm{h}^*(G) \simeq \mathrm{h}^*(G/B)/(c_1^{\mathrm{h}^*}(\mathcal{L}(\sigma(\lambda_1))),\ldots,c_1^{\mathrm{h}^*}(\mathcal{L}(\sigma(\lambda_n)))) = \mathrm{h}^*(G/B)/(\mathrm{im}(\mathfrak{c}_{G/B}|_{\mathcal{I}_F})),$$

where $\mathfrak{c}_{G/B}|_{\mathcal{I}_F}$ is the restriction of $\mathfrak{c}_{G/B}$ to $\mathcal{I}_F$. Therefore, through the identification of $\mathfrak{c}_{G/B}$ and $\bar{c}_{R[\![\Lambda^{\mathrm{sc}}]\!]_F}\circ\sigma_*$ via the isomorphism $\theta\circ\bar{\xi}^{\mathrm{sc}}$, we get

$$\begin{aligned}\mathrm{h}^*(G) \simeq \mathrm{h}^*(G/B)/(\mathrm{im}(\mathfrak{c}_{G/B}|_{\mathcal{I}_F})) &\simeq ((\mathcal{D}^{\mathrm{sc}}_F)^*/\mathcal{I}^{\mathrm{sc}}_F(\mathcal{D}^{\mathrm{sc}}_F)^*)/(\mathrm{im}((\bar{c}_{R[\![\Lambda^{\mathrm{sc}}]\!]_F}\circ\sigma_*)|_{\mathcal{I}_F})) \\ &\simeq (\mathcal{D}^*_F/\mathcal{I}_F\mathcal{D}^*_F)/(\mathrm{im}(\bar{c}_{R[\![\Lambda]\!]_F}|_{\mathcal{I}_F})) \\ &\simeq \mathcal{D}^*_F/(\mathcal{I}_F\mathcal{D}^*_F + \mathcal{C}_{R[\![\Lambda]\!]_F}).\end{aligned}$$

∎

Let $R\langle B^*\rangle$ be the free R-algebra generated by the elements $\Delta^*_{I_w}$, $w \in W$. Let $\mathcal{M}$ be the ideal in $R\langle B^*\rangle$ generated by the elements $\Delta^*_{I_w} \cdot \Delta^*_{I_{w'}} - \sum_{v\in W} \epsilon(q^v_{w,w'})\Delta^*_{I_v}$ over all $w, w' \in W$, where the $q^v_{w,w'}$ are defined after Eq. (5.2.1). Since $\epsilon\mathcal{D}^*_F$ is a free R-module with basis $\{\epsilon\Delta_{I_w}\}_{w\in W}$, we have $R\langle B^*\rangle/\mathcal{M} \simeq \mathcal{D}^*_F/\mathcal{I}_F\mathcal{D}^*_F$. Recall the $\mathbb{Z}$-basis $\{\lambda_i\}_{i=1}^n$ of Λ, and let $\mathcal{A}'$ be the ideal in $R\langle B^*\rangle/\mathcal{M}$ generated by the elements $\sum_{w\in W} \epsilon\left(\Delta_{I_w}(x_{\lambda_i})\right)\Delta^*_{I_w}$ for all $i \in [n]$. Let $\mathcal{A}$ be the ideal in $R\langle B^*\rangle$ generated by the elements $\sum_{w\in W} \epsilon\left(\Delta_{I_w}(x_{\lambda_i})\right)\Delta^*_{I_w}$ for all $i \in [n]$.

Lemma 5.3.11. *There is an isomorphism of R-algebras*

$$\mathcal{D}^*_F/(\mathcal{C}_{R[\![\Lambda]\!]_F} + \mathcal{I}_F\mathcal{D}^*_F) \simeq R\langle B^*\rangle/(\mathcal{M} + \mathcal{A}).$$

Proof: As $c_{R[\![\Lambda]\!]_F}$ is a ring homomorphism and Λ is spanned by the λ_i, $i \in [n]$, we see that $\mathcal{C}_{R[\![\Lambda]\!]_F}$ is generated by the elements $\mathrm{ev}_{x_{\lambda_i}}$, $i \in [n]$. Thus, the ideal $(\mathcal{C}_{R[\![\Lambda]\!]_F} + \mathcal{I}_F\mathcal{D}^*_F)/\mathcal{I}_F\mathcal{D}^*_F$ in $\mathcal{D}^*_F/\mathcal{I}_F\mathcal{D}^*_F$ corresponds to $\mathcal{A}'$ under the identification $\mathcal{D}^*_F/\mathcal{I}_F\mathcal{D}^*_F \simeq R\langle B^*\rangle/\mathcal{M}$. Therefore,

$$\begin{aligned}\mathcal{D}^*_F/(\mathcal{C}_{R[\![\Lambda]\!]_F} + \mathcal{I}_F\mathcal{D}^*_F) &\simeq (\mathcal{D}^*_F/\mathcal{I}_F\mathcal{D}^*_F)/\left((\mathcal{C}_{R[\![\Lambda]\!]_F} + \mathcal{I}_F\mathcal{D}^*_F)/\mathcal{I}_F\mathcal{D}^*_F\right)\\ &\simeq (R\langle B^*\rangle/\mathcal{M})/\mathcal{A}'\\ &\simeq R\langle B^*\rangle/(\mathcal{M} + \mathcal{A}).\end{aligned}$$

∎

5.4 Examples in ranks 1 and 2

In this section, we assume that G has rank 1 or 2. We compute a minimal presentation for $\mathrm{h}^*(G)$ in terms of generators and relations for several groups G.

Remark 5.4.1. Although all computations in this section are done in terms of formal Demazure operators, it may be possible to translate these computations into the language of "generalized Schubert calculus." In other words, it may be possible to use the computations performed in this section to determine the calculus of the ζ_{I_w} defined in Remark 5.3.1. If that is the case, then it should be possible to give geometric meaning to the results obtained in this section. This is potential future work.

Definition 5.4.2. *If G is simple, simply-connected (resp. adjoint), and of Dynkin type $\mathcal{D}$, let $\Lambda^{sc}_{\mathcal{D}}$ (resp. $\Lambda^{ad}_{\mathcal{D}}$) be the character lattice for T.*

Since G has rank 1 or 2, the root datum corresponding to G has rank 1 or 2, respectively. Recall that $F(x, y) = x + y + \sum_{i,j\geq 1} a_{i,j}x^iy^j \in R[\![x, y]\!]$ is a formal group law over $R = \mathrm{h}^*(\mathrm{pt})$. We summarize the results of this section in the following table:

Rank	Σ	Λ	G	$\mathrm{h}^*(G)$
1	A_1	$\Lambda^{\mathrm{sc}}_{A_1}$	$\mathrm{SL}(2,k)$	R
		$\Lambda^{\mathrm{ad}}_{A_1}$	$\mathrm{PGL}(2,k)$	$\frac{R[x]}{(2x,x^2)}$
2	$A_1 \times A_1$	$\Lambda^{\mathrm{ad}}_{A_1} \times \Lambda^{\mathrm{ad}}_{A_1}$	$\mathrm{PGL}(2,k) \times \mathrm{PGL}(2,k)$	$\frac{R[x,y]}{(2x,2y,x^2,y^2)}$
		$\Lambda^{\mathrm{ad}}_{A_1} \times \Lambda^{\mathrm{sc}}_{A_1}$	$\mathrm{PGL}(2,k) \times \mathrm{SL}(2,k)$	$\frac{R[x]}{(2x,x^2)}$
		$\Lambda^{\mathrm{sc}}_{A_1} \times \Lambda^{\mathrm{sc}}_{A_1}$	$\mathrm{SL}(2,k) \times \mathrm{SL}(2,k)$	R
2	A_2	$\Lambda^{\mathrm{ad}}_{A_2}$	$\mathrm{PGL}(3,k)$	$\frac{R[x]}{(3x,x^3)}$
		$\Lambda^{\mathrm{sc}}_{A_2}$	$\mathrm{SL}(3,k)$	R
2	B_2	$\Lambda^{\mathrm{ad}}_{B_2}$	$\mathrm{SO}(5,k)$	$\frac{R[x]}{(2x-a_{11}x^2,2x^2,x^4)}$
		$\Lambda^{\mathrm{sc}}_{B_2}$	$\mathrm{Spin}(5,k)$	R
2	G_2	$\Lambda^{\mathrm{ad}}_{G_2} = \Lambda^{\mathrm{sc}}_{G_2}$	G_2	$\frac{R[x]}{(a_{11}x,2x,x^2)}$

First we will compute the oriented cohomology rings for the rank 1 root data. Fix a simple system $\Delta = \{\alpha\}$ for Σ.

Theorem 5.4.3. *We have*

$$\mathrm{h}^*(\mathrm{PGL}(2,k)) \simeq R[x]/(2x,x^2); \qquad \mathrm{h}^*(\mathrm{SL}(2,k)) \simeq R.$$

Proof: The root system for the two groups considered in this theorem is A_1. Thus, by the computation in Example 5.2.7, the ideal $\mathcal{M}$ is generated by the following relation:

$$(1)\ \Delta^*_\alpha \Delta^*_\alpha = 0.$$

The ideal $\mathcal{A}$ depends on the character lattice Λ. We will prove this theorem case-by-case with respect to Λ.

$\underline{\Lambda = \Lambda^{\mathrm{sc}}_{A_1}}$. We have $\lambda_1 = \frac{1}{2}\alpha$. The ideal $\mathcal{A}$ is generated by the following relation:

$$(a)\ 0 = \mathrm{ev}_{x_{\lambda_1}} = \Delta^*_\alpha.$$

Therefore,

$$\mathrm{h}^*(\mathrm{SL}(2,k)) \simeq R.$$

$\underline{\Lambda = \Lambda^{\mathrm{ad}}_{A_1}}$. The ideal $\mathcal{A}$ is generated by the following relation:

$$(a)\ 0 = \mathrm{ev}_{x_\alpha} = 2\Delta^*_\alpha.$$

Therefore,

$$\mathrm{h}^*(\mathrm{PGL}(2,k)) \simeq R[x]/(2x,x^2), \quad \text{via} \quad \Delta^*_\alpha \mapsto x.$$

∎

Now we will compute that oriented cohomology rings for several rank 2 root data. Fix a basis $\{\lambda_1, \lambda_2\}$ for lattice Λ, and a simple system $\Delta = \{\alpha, \beta\}$ for the root system Σ. If Λ is simply-connected, we assume that the λ_i are the fundamental weights with respect to Δ. The Weyl group W is generated by the simple reflections s_α and s_β, and $s_\alpha s_\beta$ has order $m \in \{2,3,4,6\}$ in W. If $w \in W$ is not the longest word, then there is exactly one choice for the sequence I_w. If $w \in W$ is the longest word, then there are exactly two choices for the sequence I_w. If w is the longest word in W, we choose $I_w = \underbrace{(\alpha, \beta, \alpha, \cdots)}_{m \text{ times}}$. Thus, the set $Y = \{I_w\}_{w\in W}$ is

$$Y = \{1, \alpha, \beta, (\alpha,\beta), (\beta,\alpha), \underbrace{(\alpha,\beta,\alpha\cdots)}_{m-1 \text{ times}}, \underbrace{(\beta,\alpha,\beta,\cdots)}_{m-1 \text{ times}}, \underbrace{(\alpha,\beta,\alpha,\cdots)}_{m \text{ times}}\}.$$

The ideal $\mathcal{A}$ of Lemma 5.3.11 is generated by the elements $\sum_{w\in W} \epsilon\left(\Delta_{I_w}(x_{\lambda_1})\right) \Delta^*_{I_w}$ and $\sum_{w\in W} \epsilon\left(\Delta_{I_w}(x_{\lambda_2})\right) \Delta^*_{I_w}$. The ideal $\mathcal{M}$ is generated by the elements $\Delta^*_{I_w}\Delta^*_{I_{w'}} - \sum_{v\in W} \epsilon(q^v_{w,w'})\Delta^*_{I_v}$ over all $I_w, I_{w'}$ in Y, where the $q^v_{w,w'}$ are defined after Eq. (5.2.1).

Theorem 5.4.4. *We have*

$$\mathrm{h}^*(\mathrm{PGL}(2,k) \times \mathrm{PGL}(2,k)) \simeq R[x,y]/(2x, x^2, 2y, y^2);$$

$$\mathrm{h}^*(\mathrm{PGL}(2,k) \times \mathrm{SL}(2,k)) \simeq R[x]/(2x, x^2);$$

$$\mathrm{h}^*(\mathrm{SL}(2,k) \times \mathrm{SL}(2,k)) \simeq R.$$

Proof: The root system for all four of the groups considered in this theorem is $A_1 \times A_1$. Thus, the ideal $\mathcal{M}$ is generated by the following relations (These relations are part of the output of the code in Appendix C):

$$\begin{array}{ll} (1)\ \Delta^*_\alpha\Delta^*_\alpha = 0; & (2)\ \Delta^*_\alpha\Delta^*_\beta = \Delta^*_{(\alpha,\beta)}; \\ (3)\ \Delta^*_\alpha\Delta^*_{(\alpha,\beta)} = 0; & (4)\ \Delta^*_\beta\Delta^*_\beta = 0; \\ (5)\ \Delta^*_\beta\Delta^*_{(\alpha,\beta)} = 0; & (6)\ \Delta^*_{(\alpha,\beta)}\Delta^*_{(\alpha,\beta)} = 0. \end{array}$$

The ideal $\mathcal{A}$ depends on the character lattice Λ. We will prove this theorem case-by-case with respect to Λ.

$\underline{\Lambda = \Lambda^{\mathrm{sc}}_{A_1} \times \Lambda^{\mathrm{sc}}_{A_1}}$. We have $\lambda_1 = \frac{1}{2}\alpha$ and $\lambda_2 = \frac{1}{2}\beta$. The ideal $\mathcal{A}$ is generated by the following relations:

$$\text{(a) } 0 = \mathrm{ev}_{x_{\lambda_1}} = \Delta^*_\alpha; \quad \text{(b) } 0 = \mathrm{ev}_{x_{\lambda_2}} = \Delta^*_\beta.$$

Relations (a) and (2) imply: $\Delta^*_{(\alpha,\beta)} = 0$.

Therefore,

$$\mathrm{h}^*(\mathrm{SL}(2,k) \times \mathrm{SL}(2,k)) \simeq R.$$

$\underline{\Lambda = \Lambda^{\mathrm{ad}}_{A_1} \times \Lambda^{\mathrm{sc}}_{A_1}}$. We have $\lambda_2 = \frac{1}{2}\beta$. The ideal $\mathcal{A}$ is generated by the following relations:

$$\text{(a) } 0 = \mathrm{ev}_{x_\alpha} = 2\Delta^*_\alpha; \quad \text{(b) } 0 = \mathrm{ev}_{x_{\lambda_2}} = \Delta^*_\beta.$$

Relations (b) and (2) imply: $\Delta^*_{(\alpha,\beta)} = 0$.
Therefore,

$$\mathrm{h}^*(\mathrm{SL}(2,k) \times \mathrm{PGL}(2,k)) \simeq R[x]/(2x, x^2), \quad \text{via} \quad \Delta^*_\alpha \mapsto x.$$

$\underline{\Lambda = \Lambda^{\mathrm{ad}}_{A_1} \times \Lambda^{\mathrm{ad}}_{A_1}}$. The ideal $\mathcal{A}$ is generated by the following relations:

$$\text{(a) } 0 = \mathrm{ev}_{x_\alpha} = 2\Delta^*_\alpha; \quad \text{(b) } 0 = \mathrm{ev}_{x_\beta} = 2\Delta^*_\beta.$$

The relations imply that

$$\mathrm{h}^*(\mathrm{PGL}(2,k) \times \mathrm{PGL}(2,k)) \simeq R[x,y]/(2x, 2y, x^2, y^2), \quad \text{via} \quad \Delta^*_\alpha \mapsto x, \quad \Delta^*_\beta \mapsto y.$$

∎

Theorem 5.4.5. *We have*

$$\mathrm{h}^*(\mathrm{PGL}(3,k)) \simeq R[x]/(3x, x^3); \qquad \mathrm{h}^*(\mathrm{SL}(3,k)) \simeq R.$$

Proof: The root system for both of the groups considered in this theorem is A_2. Therefore, the ideal $\mathcal{M}$ is generated by the following relations (These relations are part of the output of the code in Appendix C):

(1) $\Delta^*_\alpha\Delta^*_\alpha = \Delta^*_{(\beta,\alpha)}$; (2) $\Delta^*_\alpha\Delta^*_\beta = \Delta^*_{(\alpha,\beta)} + \Delta^*_{(\beta,\alpha)} - a_{11}\Delta^*_{(\alpha,\beta,\alpha)}$;
(3) $\Delta^*_\alpha\Delta^*_{(\alpha,\beta)} = \Delta^*_{(\alpha,\beta,\alpha)}$; (4) $\Delta^*_\alpha\Delta^*_{(\beta,\alpha)} = 0$;
(5) $\Delta^*_\alpha\Delta^*_{(\alpha,\beta,\alpha)} = 0$; (6) $\Delta^*_\beta\Delta^*_\beta = \Delta^*_{(\alpha,\beta)}$;
(7) $\Delta^*_\beta\Delta^*_{(\alpha,\beta)} = 0$; (8) $\Delta^*_\beta\Delta^*_{(\beta,\alpha)} = \Delta^*_{(\alpha,\beta,\alpha)}$;
(9) $\Delta^*_\beta\Delta^*_{(\alpha,\beta,\alpha)} = 0$; (10) $\Delta^*_{(\alpha,\beta)}\Delta^*_{(\alpha,\beta)} = 0$;
(11) $\Delta^*_{(\alpha,\beta)}\Delta^*_{(\beta,\alpha)} = 0$; (12) $\Delta^*_{(\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha)} = 0$;
(13) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\beta,\alpha)} = 0$; (14) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)} = 0$;
(15) $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)} = 0$.

The ideal $\mathcal{A}$ depends on the character lattice Λ. We will prove this theorem case-by-case with respect to Λ.

$\underline{\Lambda = \Lambda^{\mathrm{sc}}_{A_2}}$. Choose λ_1 and λ_2 such that

$$\alpha^\vee(\lambda_1) = 1 \quad \text{and} \quad \alpha^\vee(\lambda_2) = 0 \quad \text{and} \quad \beta^\vee(\lambda_1) = 0 \quad \text{and} \quad \beta^\vee(\lambda_2) = 1.$$

Then $s_\alpha(\lambda_1) = \lambda_1 - \alpha$ and $s_\alpha(\lambda_2) = \lambda_2$ and $s_\beta(\lambda_1) = \lambda_1$ and $s_\beta(\lambda_2) = \lambda_2 - \beta$. Thus,

$$\Delta_\alpha(x_{\lambda_2}) = \Delta_\beta(x_{\lambda_1}) = 0.$$

In addition,

$$\epsilon\Delta_\alpha(x_{\lambda_1}) = \epsilon\left(\frac{x_{\lambda_1} - x_{\lambda_1 - \alpha}}{x_\alpha}\right) = \epsilon\left(\frac{x_{\lambda_1} - (x_{\lambda_1} +_F x_{-\alpha})}{x_\alpha}\right) = 1.$$

Similarly, $\epsilon\Delta_\beta(x_{\lambda_2}) = 1$. Therefore, there exist $r_{I_1}^{I_2} \in R$ such that the ideal $\mathcal{A}$ is generated by the relations

(a) $0 = \mathrm{ev}_{x_{\lambda_1}} = \Delta^*_\alpha + r_1^{(\alpha,\beta)}\Delta^*_{(\alpha,\beta)} + r_1^{(\beta,\alpha)}\Delta^*_{(\beta,\alpha)} + r_1^{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)}$,

(b) $0 = \mathrm{ev}_{x_{\lambda_2}} = \Delta^*_\beta + r_2^{(\alpha,\beta)}\Delta^*_{(\alpha,\beta)} + r_2^{(\beta,\alpha)}\Delta^*_{(\beta,\alpha)} + r_2^{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)}$.

Relations (a), (3), (10), (11), and (12) imply: (i) $\Delta^*_{(\alpha,\beta,\alpha)} = 0$.
Relations (i), (a), (1), (3), and (4) imply: (ii) $\Delta^*_{(\beta,\alpha)} = 0$.
Relations (i), (ii), (b), (6), and (7) imply: (iii) $\Delta^*_{(\alpha,\beta)} = 0$.
Relations (i), (ii), (iii), (a), and (b) imply: (iv) $\Delta^*_\alpha = \Delta^*_\beta = 0$.

By Theorem 5.3.10 and Lemma 5.3.11, the relations (a)-(b) and (1)-(15) form a complete set of relations in $\mathrm{h}^*(\mathrm{SL}(3,k))$. Therefore,

$$\mathrm{h}^*(\mathrm{SL}(3,k)) \simeq R.$$

$\underline{\Lambda = \Lambda^{\mathrm{ad}}_{A_2}}$. The ideal $\mathcal{A}$ is generated by the following relations (These relations are part of the output of the code in Appendix C):

(a) $2\Delta^*_\alpha - \Delta^*_\beta - 2a_{11}\Delta^*_{(\alpha,\beta)} + a_{11}\Delta^*_{(\beta,\alpha)} - a_{11}^2\Delta^*_{(\alpha,\beta,\alpha)} = \mathrm{ev}_{x_\alpha} = 0$,

(b) $-\Delta^*_\alpha + 2\Delta^*_\beta + a_{11}\Delta^*_{(\alpha,\beta)} - 2a_{11}\Delta^*_{(\beta,\alpha)} - (a_{11}^2 + 3a_{12})\Delta^*_{(\alpha,\beta,\alpha)} = \mathrm{ev}_{x_\beta} = 0$.

Relations (b), (3), (7), (10), (11), and (12) imply: (i) $\Delta^*_{(\alpha,\beta,\alpha)} = 0$.
Relations (i), (b), (2), (6), (7), and (8) imply: (ii) $\Delta^*_{(\alpha,\beta)} = \Delta^*_{(\beta,\alpha)}$.
Relations (i), (ii), (a), and (b) imply: (iii) $3\Delta^*_\beta = 3a_{11}\Delta^*_{(\alpha,\beta)}$.
Relations (iii), (6), and (7) imply: (iv) $3\Delta^*_\beta = 3\Delta^*_{(\alpha,\beta)} = 0$.
Relations (ii), (b), (6), and (7) imply:

(v) $\Delta^*_\beta\Delta^*_\beta\Delta^*_\beta = 0$; (vi) $\Delta^*_{(\beta,\alpha)} = \Delta^*_{(\alpha,\beta)} = \Delta^*_\beta\Delta^*_\beta$; (vii) $\Delta^*_\alpha = 2\Delta^*_\beta - a_{11}\Delta^*_\beta\Delta^*_\beta$.

In summary, $\mathrm{h}^*(\mathrm{PGL}(3,k))$ is generated as a ring by Δ^*_β, and Δ^*_β satisfies $3\Delta^*_\beta = \Delta^*_\beta\Delta^*_\beta\Delta^*_\beta = 0$. Moreover, the relations (a)-(b) and (1)-(15) are a consequence of the relations $3\Delta^*_\beta = \Delta^*_\beta\Delta^*_\beta\Delta^*_\beta = 0$. By Theorem 5.3.10 and Lemma 5.3.11, the relations (a)-(b) and (1)-(15) form a complete set of relations in $\mathrm{h}^*(\mathrm{PGL}(3,k))$. Therefore,

$$\mathrm{h}^*(\mathrm{PGL}(3,k)) \simeq R[x]/(3x, x^3), \quad \text{via} \quad \Delta^*_\beta \mapsto x.$$

∎

Theorem 5.4.6. *We have*

$$\mathrm{h}^*(\mathrm{SO}(5,k)) \simeq R[x]/(2x - a_{11}x^2, 2x^2, x^4);$$

$$\mathrm{h}^*(\mathrm{Spin}(5,k)) \simeq R.$$

Proof: The root system for both of the groups considered in this theorem is B_2. Therefore, the ideal $\mathcal{M}$ is generated by the following relations (These relations are part of the output of the code in Appendix C):

(1) $\Delta^*_\alpha \Delta^*_\alpha = \Delta^*_{(\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\alpha),(\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta)}$,

(2) $\Delta^*_\alpha \Delta^*_\beta = \Delta^*_{(\alpha,\beta)} + \Delta^*_{(\beta,\alpha)} - a_{11}\Delta^*_{(\alpha,\beta,\alpha)} - a_{11}\Delta^*_{(\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\alpha),(\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta)}$,

(3) $\Delta^*_\alpha \Delta^*_{(\alpha,\beta)} = \Delta^*_{(\alpha,\beta,\alpha)} + \Delta^*_{(\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\alpha),(\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta)}$,

(4) $\Delta^*_\alpha \Delta^*_{(\beta,\alpha)} = \Delta^*_{(\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\alpha),(\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta)}$,

(5) $\Delta^*_\alpha \Delta^*_{(\alpha,\beta,\alpha)} = 0$,

(6) $\Delta^*_\alpha \Delta^*_{(\beta,\alpha,\beta)} = \Delta^*_{(\alpha,\beta,\alpha,\beta)}$,

(7) $\Delta^*_\alpha \Delta^*_{(\alpha,\beta,\alpha,\beta)} = 0$,

(8) $\Delta^*_\beta \Delta^*_\beta = 2\Delta^*_{(\alpha,\beta)} - a_{11}\Delta^*_{(\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\beta),(\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta)}$,

(9) $\Delta^*_\beta \Delta^*_{(\alpha,\beta)} = \Delta^*_{(\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\beta),(\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta)}$,

(10) $\Delta^*_\beta \Delta^*_{(\beta,\alpha)} = 2\Delta^*_{(\alpha,\beta,\alpha)} + \Delta^*_{(\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\beta),(\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta)}$,

(11) $\Delta^*_\beta \Delta^*_{(\alpha,\beta,\alpha)} = \Delta^*_{(\alpha,\beta,\alpha,\beta)}$,

(12) $\Delta^*_\beta \Delta^*_{(\beta,\alpha,\beta)} = 0$,

(13) $\Delta^*_\beta \Delta^*_{(\alpha,\beta,\alpha,\beta)} = 0$,

(14) $\Delta^*_{(\alpha,\beta)} \Delta^*_{(\alpha,\beta)} = q^{(\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta)}$,

(15) $\Delta^*_{(\alpha,\beta)} \Delta^*_{(\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta)}$,

(16) $\Delta^*_{(\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha)} = 0$,

(17) $\Delta^*_{(\alpha,\beta)} \Delta^*_{(\beta,\alpha,\beta)} = 0$,

(18) $\Delta^*_{(\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta)} = 0$,

(19) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha),(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta)}$,

(20) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)} = 0$,

(21) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta)} = 0$,

(22) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} = 0$,

(23) $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)} = 0$,

(24) $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta)} = 0$,

(25) $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} = 0$,

(26) $\Delta^*_{(\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta)} = 0$,

(27) $\Delta^*_{(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} = 0$,

(28) $\Delta^*_{(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} = 0$.

The ideal $\mathcal{A}$ depends on the character lattice Λ. We will prove this theorem case-by-case with respect to Λ.

$\underline{\Lambda = \Lambda^{\mathrm{sc}}_{B_2}}$. Choose λ_1 and λ_2 such that

$$\alpha^\vee(\lambda_1) = 1 \quad \text{and} \quad \alpha^\vee(\lambda_2) = 0 \quad \text{and} \quad \beta^\vee(\lambda_1) = 0 \quad \text{and} \quad \beta^\vee(\lambda_2) = 1.$$

Then $s_\alpha(\lambda_1) = \lambda_1 - \alpha$ and $s_\alpha(\lambda_2) = \lambda_2$ and $s_\beta(\lambda_1) = \lambda_1$ and $s_\beta(\lambda_2) = \lambda_2 - \beta$. Thus,

$$\Delta_\alpha(x_{\lambda_2}) = \Delta_\beta(x_{\lambda_1}) = 0.$$

In addition,

$$\epsilon\Delta_\alpha(x_{\lambda_1}) = \epsilon\left(\tfrac{x_{\lambda_1} - x_{\lambda_1-\alpha}}{x_\alpha}\right) = \epsilon\left(\tfrac{x_{\lambda_1} - (x_{\lambda_1} +_F x_{-\alpha})}{x_\alpha}\right) = 1.$$

Similarly, $\epsilon\Delta_\beta(x_{\lambda_2}) = 1$. Therefore, there exist $r^{I_2}_{I_1} \in R$ such that the ideal $\mathcal{A}$ is generated by the relations

(a) $$0 = \mathrm{ev}_{x_{\lambda_1}} = \Delta^*_\alpha + r_1^{(\alpha,\beta)}\Delta^*_{(\alpha,\beta)} + r_1^{(\beta,\alpha)}\Delta^*_{(\beta,\alpha)} + r_1^{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)} + r_1^{(\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta)} + r_1^{(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)},$$

(b) $$0 = \mathrm{ev}_{x_{\lambda_2}} = \Delta^*_\beta + r_2^{(\alpha,\beta)}\Delta^*_{(\alpha,\beta)} + r_2^{(\beta,\alpha)}\Delta^*_{(\beta,\alpha)} + r_2^{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)} + r_2^{(\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta)} + r_2^{(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)}.$$

Relations (a), (6), (17), (21), (24), (26), and (27) imply: (i) $\Delta^*_{(\alpha,\beta,\alpha,\beta)} = 0$.
Relations (i), (a), (4), (15), (19), (20), and (21) imply: (ii) $\Delta^*_{(\alpha,\beta,\alpha)} = 0$.
Relations (i), (b), (9), (14), (15), (16), and (17) imply: (iii) $\Delta^*_{(\beta,\alpha,\beta)} = 0$.
Relations (i), (ii), (iii), (a), (1), (3), and (4) imply: (iv) $\Delta^*_{(\beta,\alpha)} = 0$.
Relations (i)-(iv), (a), (2), and (9) imply: (v) $\Delta^*_{(\alpha,\beta)} = 0$.
Relations (i)-(v), (a), and (b) imply: (vi) $\Delta^*_\alpha = \Delta^*_\beta = 0$.
By Theorem 5.3.10 and Lemma 5.3.11, the relations (a)-(b) and (1)-(28) form a complete set of relations in $\mathrm{h}^*(\mathrm{Spin}(5,k))$. Therefore,

$$\mathrm{h}^*(\mathrm{Spin}(5,k)) \simeq R.$$

$\underline{\Lambda = \Lambda^{\mathrm{ad}}_{B_2}}$. There are $r^{I_2}_{I_1} \in R$ such that the ideal $\mathcal{A}$ is generated by the relations (These relations are part of the output of the code in Appendix C):

$$\text{(a)}\ \ 0 = \mathrm{ev}_{x_\alpha} = 2\Delta^*_\alpha - \Delta^*_\beta - 2a_{11}\Delta^*_{(\alpha,\beta)} + a_{11}\Delta^*_{(\beta,\alpha)} - (a_{11}^2 + 2a_{12})\Delta^*_{(\beta,\alpha,\beta)} + r^{(\alpha,\beta,\alpha,\beta)}_{(\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta)},$$

$$\text{(b)}\ \ 0 = \mathrm{ev}_{x_\beta} = -2\Delta^*_\alpha + 2\Delta^*_\beta + 2a_{11}\Delta^*_{(\alpha,\beta)} - 3a_{11}\Delta^*_{(\beta,\alpha)} - 4(a_{11}^2 + a_{12})\Delta^*_{(\alpha,\beta,\alpha)} - a_{11}^2\Delta^*_{(\beta,\alpha,\beta)} + r^{(\alpha,\beta,\alpha,\beta)}_{(\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)}.$$

Relations (a), (5), (11), (16), (20), (24), and (25) imply: (i) $\Delta^*_{(\alpha,\beta,\alpha,\beta)} = 0$.
Relations (i), (a), (3), (9), (14), (15), and (17) imply: (ii) $\Delta^*_{(\beta,\alpha,\beta)} = -2\Delta^*_{(\alpha,\beta,\alpha)}$.
Relations (i), (ii), (a), (4), (10), (15), (19), and (21) imply:

$$\text{(iii)}\quad 2\Delta^*_{(\alpha,\beta,\alpha)} = -\Delta^*_{(\beta,\alpha,\beta)} = 0.$$

Relations (i), (iii), (a), and (b) imply:

$$\text{(iv)}\quad \Delta^*_\beta = 2a_{11}\Delta^*_{(\beta,\alpha)}.$$

Relations (i), (iii), (iv), (8), and (10) imply:

$$\text{(v)}\quad 2\Delta^*_{(\alpha,\beta)} = 0.$$

Relations (i), (iii), (iv), (v), (2), and (4) imply:

$$\text{(vi)}\quad \Delta^*_{(\beta,\alpha)} = \Delta^*_{(\alpha,\beta)} + a_{11}\Delta^*_{(\alpha,\beta,\alpha)}.$$

Relations (iii)-(vi) imply: (vii) $2\Delta^*_{(\beta,\alpha)} = 0$ and $\Delta^*_\beta = 0$.
Relations (i)-(vii), (b), (1), (4), and (5) imply:

(vii) $\Delta^*_{(\beta,\alpha)} = \Delta^*_\alpha\Delta^*_\alpha$; (viii) $\Delta^*_{(\alpha,\beta,\alpha)} = \Delta^*_\alpha\Delta^*_\alpha\Delta^*_\alpha$; (ix) $\Delta^*_{(\alpha,\beta)} = \Delta^*_\alpha\Delta^*_\alpha - a_{11}\Delta^*_\alpha\Delta^*_\alpha\Delta^*_\alpha$;

(x) $2\Delta^*_\alpha = a_{11}\Delta^*_\alpha\Delta^*_\alpha$; (xi) $\Delta^*_\alpha\Delta^*_\alpha\Delta^*_\alpha\Delta^*_\alpha = 0$.

In summary, $h^*(\mathrm{SO}(5,k))$ is generated as a ring by Δ^*_α, and Δ^*_α satisfies $2\Delta^*_\alpha = a_{11}\Delta^*_\alpha\Delta^*_\alpha$, $2\Delta^*_\alpha\Delta^*_\alpha = 0$, and $\Delta^*_\alpha\Delta^*_\alpha\Delta^*_\alpha\Delta^*_\alpha = 0$. Moreover, the relations (a)-(b) and (1)-(28) are a consequence of the relations $2\Delta^*_\alpha = a_{11}\Delta^*_\alpha\Delta^*_\alpha$, $2\Delta^*_\alpha\Delta^*_\alpha = 0$, and $\Delta^*_\alpha\Delta^*_\alpha\Delta^*_\alpha\Delta^*_\alpha = 0$. By Theorem 5.3.10 and Lemma 5.3.11, the relations (a)-(b) and (1)-(28) form a complete set of relations in $\mathrm{h}^*(\mathrm{SO}(5,k))$. Therefore, there is an isomorphism of rings

$$h^*(\mathrm{SO}(5,k)) \to R[x]/(x^4, 2x^2, 2x - a_{11}x^2), \quad \text{via} \quad \Delta^*_\alpha \mapsto x.$$

∎

Theorem 5.4.7. *We have*

$$\mathrm{h}^*(G_2) \simeq R[x,y]/(2x, a_{11}x, x^2).$$

Proof: There are $q^{I_3}_{I_1,I_2} \in R$ such that the ideal $\mathcal{M}$ is generated by the following relations (These relations are part of the output of the code in Appendix C):

(1) $\Delta^*_\alpha\Delta^*_\alpha = 3\Delta^*_{(\beta,\alpha)} - 3a_{11}\Delta^*_{(\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\alpha),(\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} + q^{(\beta,\alpha,\beta,\alpha)}_{(\alpha),(\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha),(\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(2) $\Delta^*_\alpha\Delta^*_\beta = \Delta^*_{(\alpha,\beta)} + \Delta^*_{(\beta,\alpha)} - a_{11}\Delta^*_{(\alpha,\beta,\alpha)} - a_{11}\Delta^*_{(\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\alpha),(\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} + q^{(\beta,\alpha,\beta,\alpha)}_{(\alpha),(\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha),(\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(3) $\Delta^*_\alpha\Delta^*_{(\alpha,\beta)} = \Delta^*_{(\alpha,\beta,\alpha)} + 3\Delta^*_{(\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\alpha),(\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} + q^{(\beta,\alpha,\beta,\alpha)}_{(\alpha),(\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha),(\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(4) $\Delta^*_\alpha\Delta^*_{(\beta,\alpha)} = 2\Delta^*_{(\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\alpha),(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} + q^{(\beta,\alpha,\beta,\alpha)}_{(\alpha),(\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha),(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(5) $\Delta^*_\alpha\Delta^*_{(\alpha,\beta,\alpha)} = 3\Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha),(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\alpha,\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(6) $\Delta^*_\alpha\Delta^*_{(\beta,\alpha,\beta)} = 2\Delta^*_{(\alpha,\beta,\alpha,\beta)} + \Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha),(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(7) $\Delta^*_\alpha\Delta^*_{(\alpha,\beta,\alpha,\beta)} = \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + 3\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(8) $\Delta^*_\alpha \Delta^*_{(\beta,\alpha,\beta,\alpha)} = \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha),(\beta,\alpha,\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(9) $\Delta^*_\alpha \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} = 0$,

(10) $\Delta^*_\alpha \Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} = \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(11) $\Delta^*_\alpha \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)} = 0$,

(12) $\Delta^*_\beta \Delta^*_\beta = \Delta^*_{(\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\beta),(\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta)} + q^{(\beta,\alpha,\beta,\alpha)}_{(\beta),(\beta)} \Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\beta),(\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\beta),(\beta)} \Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta),(\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(13) $\Delta^*_\beta \Delta^*_{(\alpha,\beta)} = 2\Delta^*_{(\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\beta),(\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta)} + q^{(\beta,\alpha,\beta,\alpha)}_{(\beta),(\alpha,\beta)} \Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\beta),(\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\beta),(\alpha,\beta)} \Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta),(\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(14) $\Delta^*_\beta \Delta^*_{(\beta,\alpha)} = \Delta^*_{(\alpha,\beta,\alpha)} + \Delta^*_{(\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta)}_{(\beta),(\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta)} + q^{(\beta,\alpha,\beta,\alpha)}_{(\beta),(\beta,\alpha)} \Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\beta),(\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\beta),(\beta,\alpha)} \Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta),(\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(15) $\Delta^*_\beta \Delta^*_{(\alpha,\beta,\alpha)} = \Delta^*_{(\alpha,\beta,\alpha,\beta)} + 2\Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\beta),(\alpha,\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\beta),(\alpha,\beta,\alpha)} \Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta),(\alpha,\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(16) $\Delta^*_\beta \Delta^*_{(\beta,\alpha,\beta)} = \Delta^*_{(\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\beta),(\beta,\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\beta),(\beta,\alpha,\beta)} \Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta),(\beta,\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(17) $\Delta^*_\beta \Delta^*_{(\alpha,\beta,\alpha,\beta)} = \Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta),(\alpha,\beta,\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(18) $\Delta^*_\beta \Delta^*_{(\beta,\alpha,\beta,\alpha)} = \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + \Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta),(\beta,\alpha,\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(19) $\Delta^*_\beta \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} = \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(20) $\Delta^*_\beta \Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} = 0$,

(21) $\Delta^*_\beta \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)} = 0$,

(22) $\Delta^*_{(\alpha,\beta)} \Delta^*_{(\alpha,\beta)} = q^{(\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta)} + q^{(\beta,\alpha,\beta,\alpha)}_{(\alpha,\beta),(\alpha,\beta)} \Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha,\beta),(\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\alpha,\beta)} \Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(23) $\Delta^*_{(\alpha,\beta)} \Delta^*_{(\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta)} + q^{(\beta,\alpha,\beta,\alpha)}_{(\alpha,\beta),(\beta,\alpha)} \Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha,\beta),(\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\beta,\alpha)} \Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(24) $\Delta^*_{(\alpha,\beta)} \Delta^*_{(\alpha,\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha,\beta),(\alpha,\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\alpha,\beta,\alpha)} \Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\alpha,\beta,\alpha)} \Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}$,

(25) $\Delta^*_{(\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta)} = q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha,\beta),(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(26) $\Delta^*_{(\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} = q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(27) $\Delta^*_{(\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta),(\beta,\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(28) $\Delta^*_{(\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} = 0,$

(29) $\Delta^*_{(\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(30) $\Delta^*_{(\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(31) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha),(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} + q^{(\beta,\alpha,\beta,\alpha)}_{(\beta,\alpha),(\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} + q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\beta,\alpha),(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha),(\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha),(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(32) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\beta,\alpha),(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha),(\alpha,\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha),(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(33) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta)} = q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\beta,\alpha),(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha),(\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha),(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(34) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} = q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha),(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(35) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha),(\beta,\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(36) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} = 0,$

(37) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(38) $\Delta^*_{(\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(39) $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha,\beta,\alpha),(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta,\alpha),(\alpha,\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta,\alpha),(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(40) $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta)} = q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha,\beta,\alpha),(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta,\alpha),(\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta,\alpha),(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(41) $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} = q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta,\alpha),(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(42) $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta,\alpha),(\beta,\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(43) $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} = 0,$

(44) $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(45) $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(46) $\Delta^*_{(\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta)} = q^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\beta,\alpha,\beta),(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + q^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha,\beta),(\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha,\beta),(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(47) $\Delta^*_{(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} = q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha,\beta),(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(48) $\Delta^*_{(\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha,\beta),(\beta,\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(49) $\Delta^*_{(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} = 0,$

(50) $\Delta^*_{(\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(51) $\Delta^*_{(\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(52) $\Delta^*_{(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} = q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta,\alpha,\beta),(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(53) $\Delta^*_{(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha,\beta,\alpha,\beta),(\beta,\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(54) $\Delta^*_{(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} = 0,$

(55) $\Delta^*_{(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(56) $\Delta^*_{(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(57) $\Delta^*_{(\beta,\alpha,\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} = q^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta,\alpha,\beta,\alpha),(\beta,\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$

(58) $\Delta^*_{(\beta,\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} = 0,$

(59) $\Delta^*_{(\beta,\alpha,\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(60) $\Delta^*_{(\beta,\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(61) $\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} = 0,$

(62) $\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(63) $\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(64) $\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(65) $\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)} = 0,$

(66) $\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)} = 0.$

There are $r^{I_2}_{I_1} \in R$ such that the ideal $\mathcal{A}$ is generated by the relations (These relations are part of the output of the code in Appendix C):

(a) $$0 = \mathrm{ev}_{x_\alpha} = 2\Delta^*_\alpha - 3\Delta^*_\beta - 3a_{11}\Delta^*_{(\alpha,\beta)} + 3a_{11}\Delta^*_{(\beta,\alpha)} - (5a_{11}^2 + a_{12})\Delta^*_{(\beta,\alpha,\beta)} + r^{(\alpha,\beta,\alpha,\beta)}_{(\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} + r^{(\beta,\alpha,\beta,\alpha)}_{(\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} + r^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + r^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + r^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)},$$

(b) $$0 = \mathrm{ev}_{x_\beta} = -\Delta^*_\alpha + 2\Delta^*_\beta + a_{11}\Delta^*_{(\alpha,\beta)} - 2a_{11}\Delta^*_{(\beta,\alpha)} - (a_{11}^2 + a_{12})\Delta^*_{(\alpha,\beta,\alpha)} + a_{11}^2\Delta^*_{(\beta,\alpha,\beta)} + r^{(\alpha,\beta,\alpha,\beta)}_{(\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} + r^{(\beta,\alpha,\beta,\alpha)}_{(\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} + r^{(\alpha,\beta,\alpha,\beta,\alpha)}_{(\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + r^{(\beta,\alpha,\beta,\alpha,\beta)}_{(\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + r^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}_{(\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}.$$

Relations (a) and (b) imply that there are $t^I \in R$ such that:

(c) $$\Delta^*_\beta = a_{11}\Delta^*_{(\alpha,\beta)} + a_{11}\Delta^*_{(\beta,\alpha)} + 2(a_{11}^2 + a_{12})\Delta^*_{(\alpha,\beta,\alpha)} + (3a_{11}^2 + a_{12})\Delta^*_{(\beta,\alpha,\beta)} + t^{(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} + t^{(\beta,\alpha,\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} + t^{(\alpha,\beta,\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + t^{(\beta,\alpha,\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + t^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}.$$

Relations (b) and (c) imply that there are $s^I \in R$ such that:

(d) $$\Delta^*_\alpha = 3a_{11}\Delta^*_{(\alpha,\beta)} + 3(a_{11}^2 + a_{12})\Delta^*_{(\alpha,\beta,\alpha)} + (7a_{11}^2 + 2a_{12})\Delta^*_{(\beta,\alpha,\beta)} + s^{(\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta)} + s^{(\beta,\alpha,\beta,\alpha)}\Delta^*_{(\beta,\alpha,\beta,\alpha)} + s^{(\alpha,\beta,\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} + s^{(\beta,\alpha,\beta,\alpha,\beta)}\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} + s^{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)}.$$

Relations (c), (19), (28), (36), (43), (49), (54), (58), (61), (62), and (63) imply:

(i) $\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha,\beta)} = 0.$

Relations (i), (c), (17), (26), (34), (41), (47), (52), (53), (54), and (55) imply:

(ii) $\Delta^*_{(\beta,\alpha,\beta,\alpha,\beta)} = 0.$

Relations (i), (ii), (c), (18), (27), (35), (42), (48), (53), (57), and (58) imply:

(iii) $\Delta^*_{(\alpha,\beta,\alpha,\beta,\alpha)} = 0.$

Relations (i), (ii), (iii), (c), (16), (25), (33), (40), (46), (47), and (48) imply:

(iv) $\Delta^*_{(\alpha,\beta,\alpha,\beta)} = 0.$

Relations (i), (ii), (iii), (iv), (d), (6), (25), (40), (46), and (48) imply:

$$\text{(v) } \Delta^*_{(\beta,\alpha,\beta,\alpha)} = 0.$$

Relations (i)-(v), (c), (14), (23), (31), (32), and (33) imply:

$$\text{(vi) } \Delta^*_{(\alpha,\beta,\alpha)} = -\Delta^*_{(\beta,\alpha,\beta)}.$$

Relations (i)-(vi), (c), (13), (22), (23), and (24) imply:

$$\text{(vii) } 2\Delta^*_{(\beta,\alpha,\beta)} = 0 \quad \text{and} \quad \Delta^*_{(\alpha,\beta,\alpha)} = \Delta^*_{(\beta,\alpha,\beta)}.$$

Relations (i)-(vii), (c), and (d) imply:

(e) $\Delta^*_\alpha = 3a_{11}\Delta^*_{(\alpha,\beta)} + a_{12}\Delta^*_{(\alpha,\beta,\alpha)}$,

(f) $\Delta^*_\beta = a_{11}\Delta^*_{(\alpha,\beta)} + a_{11}\Delta^*_{(\beta,\alpha)} + (a_{11}^2 + a_{12})\Delta^*_{(\alpha,\beta,\alpha)}$.

Relations (i)-(vii), (f), (12), (13), (14), and (15) imply:

$$\text{(viii) } \Delta^*_{(\alpha,\beta)} = 0.$$

Relations (i)-(viii), (e), (2), (13), and (15) imply:

$$\text{(ix) } \Delta^*_{(\beta,\alpha)} = -\Delta^*_{(\alpha,\beta)} = 0.$$

Relations (i)-(ix), (e), (1), and (5) imply:

$$\text{(x) } a_{11}\Delta^*_{(\alpha,\beta,\alpha)} = 0.$$

Relations (i)-(x), (e), and (f) imply:

$$\text{(xi) } \Delta^*_\alpha = \Delta^*_\beta = a_{12}\Delta^*_{(\alpha,\beta,\alpha)}.$$

Relations (i)-(xi) and (39) imply:

$$\text{(xi) } \Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)} = 0.$$

In summary, $h^*(G_2)$ is generated as a ring by $\Delta^*_{(\alpha,\beta,\alpha)}$, and $\Delta^*_{(\alpha,\beta,\alpha)}$ satisfies $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)} = 2\Delta^*_{(\alpha,\beta,\alpha)} = a_{11}\Delta^*_{(\alpha,\beta,\alpha)} = 0$. Moreover, the relations (a)-(b) and (1)-(66) are a consequence of the relations $\Delta^*_{(\alpha,\beta,\alpha)}\Delta^*_{(\alpha,\beta,\alpha)} = 2\Delta^*_{(\alpha,\beta,\alpha)} = a_{11}\Delta^*_{(\alpha,\beta,\alpha)} = 0$. By Theorem 5.3.10 and Lemma 5.3.11, the relations (a)-(b) and (1)-(66) form a complete set of relations in $\mathrm{h}^*(G_2)$. Therefore, there is an isomorphism of rings

$$\mathrm{h}^*(G_2) \to R[x]/(x^2, 2x, a_{11}x), \quad \text{via} \quad \Delta^*_{(\alpha,\beta,\alpha)} \mapsto x.$$

∎

Appendix A

Formal inverse of a formal group law

Let $F(x, y)$ be a one-dimensional commutative formal group law over a commutative, unital ring R. The purpose of this Python program is to compute the coefficients c_i of the formal inverse $g(x) = -c_1x - c_2x^2 - c_3x^3 - c_4x^4 - c_5x^5 - c_6x^6$ of x under $F(x, y)$ up to degree 6, in terms of the coefficients $a_{i,j}$ of the formal group law $F(x, y) = \sum_{i,j=1}^{\infty} a_{i,j}x^iy^j$.

```
from sympy import Symbol, poly, degree_list
from sympy.abc import x
```

The c_i are the coefficients of the formal inverse $g(x) = -c_1x - c_2x^2 - c_3x^3 - c_4x^4 - c_5x^5 - c_6x^6$.

```
c1 = Symbol('c_1')
c2 = Symbol('c_2')
c3 = Symbol('c_3')
c4 = Symbol('c_4')
c5 = Symbol('c_5')
c6 = Symbol('c_6')
```

The $a_{i,j}$ are the coefficients of the formal group law $F(x, y) = \sum_{i,j=1}^{\infty} a_{i,j}x^iy^j$ up to degree 6 terms.

```
a11 = Symbol('a_11')
a12 = Symbol('a_12')
a13 = Symbol('a_13')
a22 = Symbol('a_22')
a41 = Symbol('a_41')
a23 = Symbol('a_23')
a15 = Symbol('a_15')
a24 = Symbol('a_24')
a33 = Symbol('a_33')
```

g is the formal inverse of x under $F(x, y)$.

```
g = poly(-c1*x-c2*(x**2)-c3*(x**3)-c4*(x**4)-c5*(x**5)
    -c6*(x**6),x)
```

hij is defined to be $x^i * g(x)^j$.

```
h11 = poly(x*g,x)
h12 = poly(x*(g**2),x)
h21 = poly((x**2)*g,x)
h13 = poly(x*(g**3),x)
h22 = poly((x**2)*(g**2),x)
h31 = poly((x**3)*g,x)
h14 = poly(x*(g**4),x)
h23 = poly((x**2)*(g**3),x)
h32 = poly((x**3)*(g**2),x)
h41 = poly((x**4)*g,x)
h15 = poly(x*(g**5),x)
h24 = poly((x**2)*(g**4),x)
h33 = poly((x**3)*(g**3),x)
h42 = poly((x**4)*(g**2),x)
h51 = poly((x**5)*g,x)
```

$f_i(x)$ is a polynomial in x whose term of minimal degree is i. Since $f_i(x)$ is correct up to and including degree 6 terms, we can write $F(x, g(x)) = \sum_{i=1}^{6} f_i(x) + O(7)$, where $O(7)$ denotes a power series in x that can be divided by x^7 in $R[\![x]\!]$.

```
f1 = poly(x+g,x)
f2 = poly(a11*h11)
f3 = poly(a12*(h12+h21),x)
f4 = poly(a13*(h13+h31)+a22*(h22),x)
f5 = poly(a14*(h14+h41)+a23*(h23+h32),x)
f6 = poly(a15*(h15+h51)+a24*(h24+h42)+a33*(h33),x)
```

f is the series $F(x, g(x))$ modulo $O(7)$.

```
f = poly(f1+f2+f3+f4+f5+f6,x)
```

We collect the coefficient of x^i in $f(x)$ for $i <= 6$. Since $F(x, g(x)) = 0$ by definition, these coefficients must all be 0. This allows us to express the c_i in terms of the $a_{i,j}$.

```
for i in [1,2,3,4,5,6]:
  print("Degree", i)
  print([f.coeff_monomial(x**i)])
```

Appendix B

Coproduct coefficients

The purpose of this Python program is to compute the $p^I_{E_1,E_2}$ of Proposition 5.2.6 for reduced sequences I that lie in the set $Y = \{I_w\}$, defined in Section 5.4. Note that the $p^I_{E_1,E_2}$ for I of length 1 were computed in Example 5.2.7. Let α and β be simple roots. We provide detailed explanations for each step of this program for the reduced sequence $I_w = (\alpha, \beta, \alpha)$. The code for the other reduced sequences consisting of two simple reflections are similar, and the code is included without detailed explanations.

B.1 $I_w = (\alpha, \beta, \alpha)$

```
import numpy as np
from sympy import Symbol
```

The symbols Da and Db represent the functionals Δ^*_α and Δ^*_β, respectively. The symbols xa and xb represent the elements x^*_α and x^*_β, respectively (Here we use x^* to represent the multiplication operator in $\mathcal{D}^*_F$ corresponding to $x \in R[\![\Lambda]\!]_F$). The symbols sa and sb represent the reflections s_α and s_β across α and β, respectively. The symbols $xasa$ and $xbsb$ represent the elements $x^*_\alpha \circ s_\alpha$ and $x^*_\beta \circ s_\beta$, respectively. The symbols Aa and Bb represent the roots α and β, respectively.

```
Da = Symbol('D_a')
Db = Symbol('D_b')
xa = Symbol('x_a')
xb = Symbol('x_b')
sa = Symbol('s_a')
sb = Symbol('s_b')
xasa = Symbol('x_a*s_a')
xbsb = Symbol('x_b*s_b')
Aa = Symbol('a')
Bb = Symbol('b')
```

The function $powerset$ takes as input a sequence s. It outputs the subsequences of s, including the empty sequence. For example, if $s = [1, 2, 3]$, then

$$powerset(s) = \{[\,], [1], [2], [3], [1, 2], [1, 3], [2, 3], [1, 2, 3]\}.$$

The code we use for the powerset function is copy-pasted from Erik Cederstrand [9].

```
def powerset(s):
```

```
    x = len(s)
    masks = [1 << i for i in range(x)]
    for i in range(1 << x):
        yield [ss for mask, ss in zip(masks, s) if i & mask]
```

X is the empty sequence. X will store elements of the form [[S1,S2],[B1,B2,B3]].

```
X = []
```

Let E_1 and E_2 be subsequences of $[1,2,3]$ (Note that specifying a subset of $[1,2,3]$ is the same as specifying a subsequence of $[1,2,3]$). The following lines of code use the algorithm defined in Proposition 5.2.6 to compute the $p^I_{E_1,E_2}$ of Proposition 5.2.6. In particular, this code computes the B_j, $j = 1,2,3$, with respect to E_1 and E_2. It then defines the sequences S1 and S2, which equal the sequences $I|_{E_1}$ and $I|_{E_2}$ of Proposition 5.2.6. Finally, it appends the element [[S1,S2],[B1,B2,B3]] to X. Observe that $p^I_{E_1,E_2} = B_1 \circ B_2 \circ B_3(1)$. This process is repeated for all pairs (E_1, E_2) using a nested for-loop.

```
for E1 in powerset([1,2,3]):
    for E2 in powerset([1,2,3]):
        if (1 in E1) & (1 in E2):
            B1 = -xasa
        elif (1 not in E1) & (1 not in E2):
            B1 = Da
        else:
            B1 = sa
        if (2 in E1) & (2 in E2):
            B2 = -xbsb
        elif (2 not in E1) & (2 not in E2):
            B2 = Db
        else:
            B2 = sb
        if (3 in E1) & (3 in E2):
            B3 = -xasa
        elif (3 not in E1) & (3 not in E2):
            B3 = Da
        else:
            B3 = sa
        S1 = [];
        S2 = [];
        for x in E1:
             if (x == 1) | (x == 3):
                  S1.append(Aa)
             elif (x == 2):
                  S1.append(Bb)
        for y in E2:
             if (y == 1) | (y == 3):
                 S2.append(Aa)
            elif (y == 2):
                 S2.append(Bb)
        X.append([[S1,S2],[B1,B2,B3]])
```

We define several variables.

```
X0 = []
sab = {xasa,xbsb}
Dab = {Da,Db}
b = 0
```

The following code tags elements $[[E1, E2], [B1, B2, B3]]$ of X such that $p^I_{E_1,E_2} = B_1 \circ B_2 \circ B_3(1) = 0 \bmod \mathcal{I}_F\mathcal{D}^*_F$. The elements not tagged are stored in the sequence $X0$.

```
for x in X:
    if ((x[1][2] == Da) | (x[1][2] == Db)):
```

```
            b = 1
        elif ((x[1][1] == Da) | (x[1][1] == Db)) & (-x[1][2] not in sab):
            b = 1
        elif ((x[1][0] == Da) | (x[1][0] == Db)) & ((-x[1][1] not in sab) & (-x[1][2] not in sab)):
            b = 1
        elif (-x[1][0] in sab):
            b = 1
        elif ((x[1][0] not in Dab) & (x[1][1] not in Dab) & (x[1][2] not in Dab))
        & ((-x[1][0] in sab) | (-x[1][1] in sab) | (-x[1][2] in sab)):
            b = 1
        if b == 0:
            X0.append(x)
        b = 0
```

Y is the empty sequence. It will store the elements of powerset([Aa,Bb,Aa]). The following code appends the elements of powerset([Aa,Bb,Aa]) to Y. Note that the elements of Y will not necessarily be unique. For example, [Aa] will appear twice in the sequence Y.

```
    Y = []
    for E1 in powerset([Aa,Bb,Aa]):
        Y.append(E1)
```

Z is the empty sequence. It will store the elements of Y, without repeated elements. The following code tags elements of Y that have not yet been tagged, and appends these elements to Z.

```
    Z = []
    for y in Y:
        if y not in Z:
            Z.append(y)
```

ZZ1 is the empty sequence. It stores pairs of elements of Z.

```
    ZZ1 = []
    for z1 in Z:
        for z2 in Z:
            ZZ1.append([z1,z2])
```

ZZ is the empty sequence. It will store elements in ZZ1, minus repeated elements.

```
    ZZ = []
    for zz in ZZ1:
        if zz not in ZZ:
            ZZ.append(zz)
```

Let [Z1,Z2] be an element in ZZ. Then R contains elements in ZZ such that either Xa or Xb appears consecutively in Z1 or Z2.

```
    R = []
    for zz in ZZ:
        if zz[0] != []:
            l = len(zz[0])
            for i in range(l-1):
                if zz[0][i] == zz[0][i+1]:
                    R.append(zz)
    for zz in ZZ:
        if zz[1] != []:
            l = len(zz[1])
            for i in range(l-1):
                if zz[1][i] == zz[1][i+1]:
                    R.append(zz)
```

RR will contain elements in R, minus repeated elements.

```
RR = []
for r in R:
     if r not in RR:
          RR.append(r)
```

Let [Z1,Z2] be an element in ZZ. Then SS contains elements in ZZ such that *neither* Xa nor Xb appears consecutively in Z1 and Z2.

```
SS = []
for zz in ZZ:
     if zz not in RR:
          SS.append(zz)
```

L1 will store pairs [ss,t1], where ss=[S1,S2] equals x[0] for some x in X0 such that neither S1 nor S2 contains Xa consecutively nor Xb consecutively, and t1 consists of the elements x[1] such that x is in X0 and x[0]=ss. The sum over the elements of t1 is part of the coefficient of the coproduct of XaXbXa at the tensor product of the pair of elements in ss (the remaining part of the coproduct coefficient can be computed using L2 defined below).

```
L1 = []
t1 = []
for ss in SS:
     for x in X0:
          if (x[0] == ss):
               t1.append(x[1])
     L1.append([ss,t1])
     t1 = []
print(*L1, sep = "\n")
```

L2 will store pairs [rr,t2], where rr=[R1,R2] equals x[0] for some x in X0 such that either R1 or R2 contains Xa or Xb consecutively, and t2 consists of the elements x[1] such that x is in X0 and x[0]=rr.

```
L2 = []
t2 = []
for rr in RR:
     for x in X0:
          if (x[0] == rr):
               t2.append(x[1])
     L2.append([rr,t2])
     t2 = []
print(*L2, sep = "\n")
```

B.2 $I_w = (\alpha, \beta)$

```
import numpy as np
from sympy import Symbol

Da = Symbol('D_a')
Db = Symbol('D_b')
xa = Symbol('x_a')
xb = Symbol('x_b')
sa = Symbol('s_a')
sb = Symbol('s_b')
xasa = Symbol('x_a*s_a')
xbsb = Symbol('x_b*s_b')
Aa = Symbol('a')
Bb = Symbol('b')
```

```
def powerset(s):
x = len(s)
masks = [1 << i for i in range(x)]
for i in range(1 << x):
     yield [ss for mask, ss in zip(masks, s) if i & mask]

X = []
for E1 in powerset([1,2]):
     for E2 in powerset([1,2]):
          if (1 in E1) & (1 in E2):
               B1 = -xasa
          elif (1 not in E1) & (1 not in E2):
               B1 = Da
          else:
               B1 = sa
          if (2 in E1) & (2 in E2):
               B2 = -xbsb
          elif (2 not in E1) & (2 not in E2):
               B2 = Db
          else:
               B2 = sb
          S1 = [];
          S2 = [];
          for x in E1:
                if (x == 1):
                     S1.append(Aa)
                elif (x == 2):
                     S1.append(Bb)
          for y in E2:
                if (y == 1):
                     S2.append(Aa)
               elif (y == 2):
                     S2.append(Bb)
          X.append([[S1,S2],[B1,B2]])

X0 = []
sab = {xasa,xbsb}
Dab = {Da,Db}
b = 0

for x in X:
     if ((x[1][1] == Da) | (x[1][1] == Db)):
          b = 1
     elif ((x[1][0] == Da) | (x[1][0] == Db)) & (-x[1][1] not in sab):
          b = 1
     elif ((x[1][0] not in Dab) & (x[1][1] not in Dab)) & ((-x[1][0] in sab) | (-x[1][1] in sab)):
          b = 1
     if b == 0:
          X0.append(x)
     b = 0

Y = []
for E1 in powerset([Aa,Bb]):
     Y.append(E1)
```

The rest of this code is the same as for $I_w = (\alpha, \beta, \alpha)$.

B.3 $I_w = (\beta, \alpha)$

```
import numpy as np
from sympy import Symbol

Da = Symbol('D_a')
Db = Symbol('D_b')
xa = Symbol('x_a')
xb = Symbol('x_b')
sa = Symbol('s_a')
sb = Symbol('s_b')
xasa = Symbol('x_a*s_a')
xbsb = Symbol('x_b*s_b')
Aa = Symbol('a')
Bb = Symbol('b')

def powerset(s):
x = len(s)
masks = [1 << i for i in range(x)]
for i in range(1 << x):
     yield [ss for mask, ss in zip(masks, s) if i & mask]

X = []

for E1 in powerset([1,2]):
     for E2 in powerset([1,2]):
          if (1 in E1) & (1 in E2):
               B1 = -xbsb
          elif (1 not in E1) & (1 not in E2):
               B1 = Db
          else:
               B1 = sb
          if (2 in E1) & (2 in E2):
               B2 = -xasa
          elif (2 not in E1) & (2 not in E2):
               B2 = Da
          else:
               B2 = sa
          S1 = [];
          S2 = [];
          for x in E1:
                if (x == 1):
                     S1.append(Bb)
                elif (x == 2):
                     S1.append(Aa)
          for y in E2:
                if (y == 1):
                    S2.append(Bb)
               elif (y == 2):
                    S2.append(Aa)
          X.append([[S1,S2],[B1,B2]])

X0 = []
sab = {xasa,xbsb}
Dab = {Da,Db}
b = 0

for x in X:
     if ((x[1][1] == Da) | (x[1][1] == Db)):
          b = 1
     elif ((x[1][0] == Da) | (x[1][0] == Db)) & (-x[1][1] not in sab):
          b = 1
```

```
        elif ((x[1][0] not in Dab) & (x[1][1] not in Dab)) & ((-x[1][0] in sab) | (-x[1][1] in sab)):
            b = 1
        if b == 0:
            X0.append(x)
        b = 0

Y = []
for E1 in powerset([Bb,Aa]):
    Y.append(E1)
```

The rest of this code is the same as for $I_w = (\alpha, \beta, \alpha)$.

B.4 $I_w = (\beta, \alpha, \beta)$

```
import numpy as np
from sympy import Symbol

Da = Symbol(‘D_a’)
Db = Symbol(‘D_b’)
xa = Symbol(‘x_a’)
xb = Symbol(‘x_b’)
sa = Symbol(‘s_a’)
sb = Symbol(‘s_b’)
xasa = Symbol(‘x_a*s_a’)
xbsb = Symbol(‘x_b*s_b’)
Aa = Symbol(‘a’)
Bb = Symbol(‘b’)

def powerset(s):
x = len(s)
masks = [1 << i for i in range(x)]
for i in range(1 << x):
    yield [ss for mask, ss in zip(masks, s) if i & mask]

X = []

for E1 in powerset([1,2,3]):
    for E2 in powerset([1,2,3]):
        if (1 in E1) & (1 in E2):
            B1 = -xbsb
        elif (1 not in E1) & (1 not in E2):
            B1 = Db
        else:
            B1 = sb
        if (2 in E1) & (2 in E2):
            B2 = -xasa
        elif (2 not in E1) & (2 not in E2):
            B2 = Da
        else:
            B2 = sa
        if (3 in E1) & (3 in E2):
            B3 = -xbsb
        elif (3 not in E1) & (3 not in E2):
            B3 = Db
        else:
            B3 = sb
        S1 = [];
        S2 = [];
        for x in E1:
```

```
                if (x == 1) | (x == 3):
                     S1.append(Bb)
                elif (x == 2):
                     S1.append(Aa)
           for y in E2:
                if (y == 1) | (y == 3):
                     S2.append(Bb)
                elif (y == 2):
                     S2.append(Aa)
           X.append([[S1,S2],[B1,B2,B3]])
X0 = []
sab = {xasa,xbsb}
Dab = {Da,Db}
b = 0

for x in X:
      if ((x[1][2] == Da) | (x[1][2] == Db)):
           b = 1
      elif ((x[1][1] == Da) | (x[1][1] == Db)) & (-x[1][2] not in sab):
           b = 1
      elif ((x[1][0] == Da) | (x[1][0] == Db)) & ((-x[1][1] not in sab)
      & (-x[1][2] not in sab)):
           b = 1
      elif (-x[1][0] in sab):
           b = 1
      elif ((x[1][0] not in Dab) & (x[1][1] not in Dab) & (x[1][2] not in Dab))
      & ((-x[1][0] in sab) | (-x[1][1] in sab) | (-x[1][2] in sab)):
           b = 1
      if b == 0:
           X0.append(x)
      b = 0

Y = []

for E1 in powerset([Bb,Aa,Bb]):
      Y.append(E1)
```

The rest of this code is the same as for $I_w = (\alpha, \beta, \alpha)$.

B.5 $I_w = (\alpha, \beta, \alpha, \beta)$

```
import numpy as np
from sympy import Symbol

Da = Symbol('D_a')
Db = Symbol('D_b')
xa = Symbol('x_a')
xb = Symbol('x_b')
sa = Symbol('s_a')
sb = Symbol('s_b')
xasa = Symbol('x_a*s_a')
xbsb = Symbol('x_b*s_b')
Aa = Symbol('a')
Bb = Symbol('b')

def powerset(s):
x = len(s)
masks = [1 << i for i in range(x)]
for i in range(1 << x):
```

```
        yield [ss for mask, ss in zip(masks, s) if i & mask]

X = []

for E1 in powerset([1,2,3,4]):
    for E2 in powerset([1,2,3,4]):
        if (1 in E1) & (1 in E2):
            B1 = -xasa
        elif (1 not in E1) & (1 not in E2):
            B1 = Da
        else:
            B1 = sa
        if (2 in E1) & (2 in E2):
            B2 = -xbsb
        elif (2 not in E1) & (2 not in E2):
            B2 = Db
        else:
            B2 = sb
        if (3 in E1) & (3 in E2):
            B3 = -xasa
        elif (3 not in E1) & (3 not in E2):
            B3 = Da
        else:
            B3 = sa
        if (4 in E1) & (4 in E2):
            B4 = -xbsb
        elif (4 not in E1) & (4 not in E2):
            B4 = Db
        else:
            B4 = sb
        S1 = [];
        S2 = [];
        for x in E1:
            if (x == 1) | (x == 3):
                S1.append(Aa)
            elif (x == 2) | (x == 4):
                S1.append(Bb)
        for y in E2:
            if (y == 1) | (y == 3):
                S2.append(Aa)
            elif (y == 2) | (y == 4):
                S2.append(Bb)
        X.append([[S1,S2],[B1,B2,B3,B4]])

X0 = []
sab = {xasa,xbsb}
Dab = {Da,Db}
b = 0

for x in X:
    if ((x[1][3] == Da) | (x[1][3] == Db)):
        b = 1
    elif ((x[1][2] == Da) | (x[1][2] == Db)) & (-x[1][3] not in sab):
        b = 1
    elif ((x[1][1] == Da) | (x[1][1] == Db)) & ((-x[1][2] not in sab) & (-x[1][3] not in sab)):
        b = 1
    elif ((x[1][0] == Da) | (x[1][0] == Db)) & ((-x[1][1] not in sab) & (-x[1][2] not in sab)
    & (-x[1][3] not in sab)):
        b = 1
    elif (-x[1][0] in sab):
        b = 1
```

```
        elif ((x[1][0] not in Dab) & (x[1][1] not in Dab) & (x[1][2] not in Dab)
        & (x[1][3] not in Dab)) & ((-x[1][0] in sab) | (-x[1][1] in sab) | (-x[1][2] in sab)
        | (-x[1][3] in sab)):
            b = 1
        if b == 0:
            X0.append(x)
            b = 0

Y = []
for E1 in powerset([Aa,Bb,Aa,Bb]):
    Y.append(E1)
# The rest of this code is the same for I_w = (α, β, α).
```

B.6 $I_w = (\beta, \alpha, \beta, \alpha)$

```
import numpy as np
from sympy import Symbol

Da = Symbol('D_a')
Db = Symbol('D_b')
xa = Symbol('x_a')
xb = Symbol('x_b')
sa = Symbol('s_a')
sb = Symbol('s_b')
xasa = Symbol('x_a*s_a')
xbsb = Symbol('x_b*s_b')
Aa = Symbol('a')
Bb = Symbol('b')

def powerset(s):
x = len(s)
masks = [1 << i for i in range(x)]
for i in range(1 << x):
    yield [ss for mask, ss in zip(masks, s) if i & mask]

X = []
for E1 in powerset([1,2,3,4]):
    for E2 in powerset([1,2,3,4]):
        if (1 in E1) & (1 in E2):
            B1 = -xbsb
        elif (1 not in E1) & (1 not in E2):
            B1 = Db
        else:
            B1 = sb
        if (2 in E1) & (2 in E2):
            B2 = -xasa
        elif (2 not in E1) & (2 not in E2):
            B2 = Da
        else:
            B2 = sa
        if (3 in E1) & (3 in E2):
            B3 = -xbsb
        elif (3 not in E1) & (3 not in E2):
            B3 = Db
         else:
            B3 = sb
        if (4 in E1) & (4 in E2):
```

```
                B4 = -xasa
            elif (4 not in E1) & (4 not in E2):
                B4 = Da
            else:
                B4 = sa
            S1 = [];
            S2 = [];
            for x in E1:
                if (x == 1) | (x == 3):
                    S1.append(Bb)
                elif (x == 2) | (x == 4):
                    S1.append(Aa)
            for y in E2:
                if (y == 1) | (y == 3):
                    S2.append(Bb)
                elif (y == 2) | (y == 4):
                    S2.append(Aa)
            X.append([[S1,S2],[B1,B2,B3,B4]])

X0 = []
sab = {xasa,xbsb}
Dab = {Da,Db}
b = 0

for x in X:
    if ((x[1][3] == Da) | (x[1][3] == Db)):
        b = 1
    elif ((x[1][2] == Da) | (x[1][2] == Db)) & (-x[1][3] not in sab):
        b = 1
    elif ((x[1][1] == Da) | (x[1][1] == Db)) & ((-x[1][2] not in sab) & (-x[1][3] not in sab)):
        b = 1
    elif ((x[1][0] == Da) | (x[1][0] == Db)) & ((-x[1][1] not in sab) & (-x[1][2] not in sab)
    & (-x[1][3] not in sab)):
        b = 1
    elif (-x[1][0] in sab):
        b = 1
    elif ((x[1][0] not in Dab) & (x[1][1] not in Dab) & (x[1][2] not in Dab)
    & (x[1][3] not in Dab)) & ((-x[1][0] in sab) | (-x[1][1] in sab) | (-x[1][2] in sab)
    | (-x[1][3] in sab)):
        b = 1
    if b == 0:
        X0.append(x)
        b = 0

Y = []
for E1 in powerset([Bb,Aa,Bb,Aa]):
    Y.append(E1)
```

The rest of this code is the same as for $I_w = (\alpha, \beta, \alpha)$.

B.7 $I_w = (\alpha, \beta, \alpha, \beta, \alpha)$

```
import numpy as np
from sympy import Symbol

Da = Symbol('D_a')
Db = Symbol('D_b')
xa = Symbol('x_a')
xb = Symbol('x_b')
```

```
sa = Symbol('s_a')
sb = Symbol('s_b')
xasa = Symbol('x_a*s_a')
xbsb = Symbol('x_b*s_b')
Aa = Symbol('a')
Bb = Symbol('b')

def powerset(s):
x = len(s)
masks = [1 << i for i in range(x)]
for i in range(1 << x):
     yield [ss for mask, ss in zip(masks, s) if i & mask]

X = []
for E1 in powerset([1,2,3,4,5]):
     for E2 in powerset([1,2,3,4,5]):
          if (1 in E1) & (1 in E2):
               B1 = -xasa
          elif (1 not in E1) & (1 not in E2):
               B1 = Da
          else:
               B1 = sa
          if (2 in E1) & (2 in E2):
               B2 = -xbsb
          elif (2 not in E1) & (2 not in E2):
               B2 = Db
          else:
               B2 = sb
          if (3 in E1) & (3 in E2):
               B3 = -xasa
          elif (3 not in E1) & (3 not in E2):
               B3 = Da
           else:
               B3 = sa
          if (4 in E1) & (4 in E2):
               B4 = -xbsb
          elif (4 not in E1) & (4 not in E2):
               B4 = Db
          else:
               B4 = sb
          if (5 in E1) & (5 in E2):
               B5 = -xasa
          elif (5 not in E1) & (5 not in E2):
               B5 = Da
          else:
               B5 = sa
          S1 = [];
          S2 = [];
          for x in E1:
               if (x == 1) | (x == 3) | (x == 5):
                    S1.append(Aa)
               elif (x == 2) | (x == 4):
                    S1.append(Bb)
          for y in E2:
               if (y == 1) | (y == 3) | (x == 5):
                    S2.append(Aa)
               elif (y == 2) | (y == 4):
                    S2.append(Bb)
          X.append([[S1,S2],[B1,B2,B3,B4,B5]])

X0 = []
sab = {xasa,xbsb}
```

```
Dab = {Da,Db}
b = 0

for x in X:
      if ((x[1][4] == Da) | (x[1][4] == Db)):
            b = 1
      elif ((x[1][3] == Da) | (x[1][3] == Db)) & (-x[1][4] not in sab):
            b = 1
      elif ((x[1][2] == Da) | (x[1][2] == Db)) & ((-x[1][3] not in sab) & (-x[1][4] not in sab)):
            b = 1
      elif ((x[1][1] == Da) | (x[1][1] == Db)) & ((-x[1][2] not in sab) & (-x[1][3] not in sab)
      & (-x[1][4] not in sab)):
            b = 1
      elif ((x[1][0] == Da) | (x[1][0] == Db)) & ((-x[1][1] not in sab) & (-x[1][2] not in sab)
      & (-x[1][3] not in sab) & (-x[1][4] not in sab)):
            b = 1
      elif (-x[1][0] in sab):
            b = 1
      elif ((x[1][0] not in Dab) & (x[1][1] not in Dab) & (x[1][2] not in Dab)
      & (x[1][3] not in Dab) & (x[1][4] not in Dab)) & ((-x[1][0] in sab) | (-x[1][1] in sab)
      | (-x[1][2] in sab) | (-x[1][3] in sab) | (-x[1][4] in sab)):
            b = 1
      if b == 0:
            X0.append(x)
            b = 0

Y = []
for E1 in powerset([Aa,Bb,Aa,Bb,Aa]):
      Y.append(E1)
```

The rest of this code is the same as for $I_w = (\alpha, \beta, \alpha)$.

B.8 $I_w = (\beta, \alpha, \beta, \alpha, \beta)$

```
import numpy as np
from sympy import Symbol

Da = Symbol('D_a')
Db = Symbol('D_b')
xa = Symbol('x_a')
xb = Symbol('x_b')
sa = Symbol('s_a')
sb = Symbol('s_b')
xasa = Symbol('x_a*s_a')
xbsb = Symbol('x_b*s_b')
Aa = Symbol('a')
Bb = Symbol('b')

def powerset(s):
x = len(s)
masks = [1 << i for i in range(x)]
for i in range(1 << x):
      yield [ss for mask, ss in zip(masks, s) if i & mask]

X = []
for E1 in powerset([1,2,3,4,5]):
      for E2 in powerset([1,2,3,4,5]):
            if (1 in E1) & (1 in E2):
                  B1 = -xbsb
```

```
            elif (1 not in E1) & (1 not in E2):
                 B1 = Db
            else:
                 B1 = sb
            if (2 in E1) & (2 in E2):
                 B2 = -xasa
            elif (2 not in E1) & (2 not in E2):
                 B2 = Da
            else:
                 B2 = sa
            if (3 in E1) & (3 in E2):
                 B3 = -xbsb
            elif (3 not in E1) & (3 not in E2):
                 B3 = Db
            else:
                 B3 = sb
            if (4 in E1) & (4 in E2):
                 B4 = -xasa
            elif (4 not in E1) & (4 not in E2):
                 B4 = Da
            else:
                 B4 = sa
            if (5 in E1) & (5 in E2):
                 B5 = -xbsb
            elif (5 not in E1) & (5 not in E2):
                 B5 = Db
            else:
                 B5 = sb
            S1 = [];
            S2 = [];
            for x in E1:
                 if (x == 1) | (x == 3) | (x == 5):
                      S1.append(Bb)
                 elif (x == 2) | (x == 4):
                      S1.append(Aa)
            for y in E2:
                 if (y == 1) | (y == 3) | (x == 5):
                      S2.append(Bb)
                 elif (y == 2) | (y == 4):
                      S2.append(Aa)
            X.append([[S1,S2],[B1,B2,B3,B4,B5]])

X0 = []
sab = {xasa,xbsb}
Dab = {Da,Db}
b = 0
for x in X:
      if ((x[1][4] == Da) | (x[1][4] == Db)):
            b = 1
      elif ((x[1][3] == Da) | (x[1][3] == Db)) & (-x[1][4] not in sab):
            b = 1
      elif ((x[1][2] == Da) | (x[1][2] == Db)) & ((-x[1][3] not in sab) & (-x[1][4] not in sab)):
            b = 1
      elif ((x[1][1] == Da) | (x[1][1] == Db)) & ((-x[1][2] not in sab) & (-x[1][3] not in sab)
      & (-x[1][4] not in sab)):
            b = 1
      elif ((x[1][0] == Da) | (x[1][0] == Db)) & ((-x[1][1] not in sab) & (-x[1][2] not in sab)
      & (-x[1][3] not in sab) & (-x[1][4] not in sab)):
            b = 1
      elif (-x[1][0] in sab):
            b = 1
```

```
        elif ((x[1][0] not in Dab) & (x[1][1] not in Dab) & (x[1][2] not in Dab)
        & (x[1][3] not in Dab) & (x[1][4] not in Dab)) & ((-x[1][0] in sab) | (-x[1][1] in sab)
        | (-x[1][2] in sab) | (-x[1][3] in sab) | (-x[1][4] in sab)):
            b = 1
        if b == 0:
            X0.append(x)
            b = 0

Y = []
for E1 in powerset([Bb,Aa,Bb,Aa,Bb]):
    Y.append(E1)
# The rest of this code is the same as for I_w = (α, β, α).
```

B.9 $I_w = (\alpha, \beta, \alpha, \beta, \alpha, \beta)$

```
import numpy as np
from sympy import Symbol

Da = Symbol('D_a')
Db = Symbol('D_b')
xa = Symbol('x_a')
xb = Symbol('x_b')
sa = Symbol('s_a')
sb = Symbol('s_b')
xasa = Symbol('x_a*s_a')
xbsb = Symbol('x_b*s_b')
Aa = Symbol('a')
Bb = Symbol('b')

def powerset(s):
x = len(s)
masks = [1 << i for i in range(x)]
for i in range(1 << x):
    yield [ss for mask, ss in zip(masks, s) if i & mask]

X = []
for E1 in powerset([1,2,3,4,5,6]):
    for E2 in powerset([1,2,3,4,5,6]):
        if (1 in E1) & (1 in E2):
            B1 = -xasa
        elif (1 not in E1) & (1 not in E2):
            B1 = Da
        else:
            B1 = sa
        if (2 in E1) & (2 in E2):
            B2 = -xbsb
        elif (2 not in E1) & (2 not in E2):
            B2 = Db
        else:
            B2 = sb
        if (3 in E1) & (3 in E2):
            B3 = -xasa
        elif (3 not in E1) & (3 not in E2):
            B3 = Da
        else:
            B3 = sa
        if (4 in E1) & (4 in E2):
            B4 = -xbsb
```

```
            elif (4 not in E1) & (4 not in E2):
                B4 = Db
            else:
                B4 = sb
            if (5 in E1) & (5 in E2):
                B5 = -xasa
            elif (5 not in E1) & (5 not in E2):
                B5 = Da
            else:
                B5 = sa
            if (6 in E1) & (6 in E2):
                B6 = -xbsb
            elif (6 not in E1) & (6 not in E2):
                B6 = Db
            else:
                B6 = sb
            S1 = [];
            S2 = [];
            for x in E1:
                if (x == 1) | (x == 3) | (x == 5):
                    S1.append(Aa)
                elif (x == 2) | (x == 4) | (x == 6):
                    S1.append(Bb)
            for y in E2:
                if (y == 1) | (y == 3) | (x == 5):
                    S2.append(Aa)
                elif (y == 2) | (y == 4) | (x == 6):
                    S2.append(Bb)
            X.append([[S1,S2],[B1,B2,B3,B4,B5,B6]])

    X0 = []
    sab = {xasa,xbsb}
    Dab = {Da,Db}
    b = 0

    for x in X:
        if (x[1][5] == Da) | (x[1][5] == Db):
            b = 1
        elif ((x[1][4] == Da) | (x[1][4] == Db)) & (-x[1][5] not in sab):
            b = 1
        elif ((x[1][3] == Da) | (x[1][3] == Db)) & ((-x[1][4] not in sab) & (-x[1][5] not in sab)):
            b = 1
        elif ((x[1][2] == Da) | (x[1][2] == Db)) & ((-x[1][3] not in sab) & (-x[1][4] not in sab) &
        (-x[1][5] not in sab)):
            b = 1
        elif ((x[1][1] == Da) | (x[1][1] == Db)) & ((-x[1][2] not in sab) & (-x[1][3] not in sab) &
        (-x[1][4] not in sab) & (-x[1][5] not in sab)):
            b = 1
        elif ((x[1][0] == Da) | (x[1][0] == Db)) & ((-x[1][1] not in sab) & (-x[1][2] not in sab)
        & (-x[1][3] not in sab) & (-x[1][4] not in sab) & (-x[1][5] not in sab)):
            b = 1
        elif (-x[1][0] in sab):
            b = 1
        elif ((x[1][0] not in Dab) & (x[1][1] not in Dab) & (x[1][2] not in Dab) & (x[1][3] not in Dab)
        & (x[1][4] not in Dab) & (x[1][5] not in Dab)) & ((-x[1][0] in sab) | (-x[1][1] in sab) |
        (-x[1][2] in sab) | (-x[1][3] in sab) | (-x[1][4] in sab) | (-x[1][5] in sab)):
            b = 1
        if b == 0:
            X0.append(x)
            b = 0

    Y = []
```

```
for E1 in powerset([Aa,Bb,Aa,Bb,Aa,Bb]):
     Y.append(E1)
```
The rest of this code is the same as for $I_w = (\alpha, \beta, \alpha)$.

Appendix C

Complete set of relations for the oriented cohomology rings of the semisimple adjoint groups of rank 2

Let $\{\alpha, \beta\}$ be simple roots for an adjoint root datum. This Python program computes ev_α and ev_β in $\mathcal{D}_F^*$ for the adjoint groups of rank 2. It also computes the $q^v_{w,w'}$ of Equation 5.2.1. We include the code for all root systems of rank 2, with detailed explanations of each step for the root system A_2.

C.1 A_2

```
from sympy import Symbol, poly, div, degree_list
from sympy.abc import x, y
```

The xa and xb are the elements x_α and x_β in the formal group algebra, respectively. The elements Xa, Xb, Xab, Xba, and Xaba represent the functionals Δ_α^*, Δ_β^*, $(\Delta_\alpha\Delta_\beta)^*$, $(\Delta_\beta\Delta_\alpha)^*$, and $(\Delta_\alpha\Delta_\beta\Delta_\alpha)^*$, respectively. The aij are the coefficients $a_{i,j}$ that appear in the formal group law $F(x,y) = x + y + \sum_{i,j\geq 1} a_{i,j}x^iy^i$. The cj are the coefficients c_j that appear in the formal inverse $-_Fx = -x - c_1x - c_2x^2 - \cdots$ of x under the formal group law F. The symbols x_nega and x_negb represent the elements $x_{-\alpha}$ and $x_{-\beta}$ in the formal group algebra, respectively, truncated to the degree 3 terms. The symbol xab represents the element $x_{\alpha+\beta}$ in the formal group algebra, truncated to the degree 3. The symbols saxb and sbxa represent the elements $s_\alpha(x_\beta)$ and $s_\beta(x_\alpha)$ in the formal group algebra, respectively, truncated to the degree 3 terms. The symbols ka and kb represent the constant terms of the elements $\kappa_\alpha = \frac{1}{x_\alpha} + \frac{1}{x_{-\alpha}}$ and $\kappa_\beta = \frac{1}{x_\beta} + \frac{1}{x_{-\beta}}$. The constant term of both κ_α and κ_β is $-a_{11}$.

```
xa = Symbol('x_a')
xb = Symbol('x_b')
Xa = Symbol('X_a')
Xb = Symbol('X_b')
Xab = Symbol('X_ab')
Xba = Symbol('X_ba')
Xaba = Symbol('X_aba')
a11 = Symbol('a_11')
a12 = Symbol('a_12')
c2 = -a11
c3 = -a11*c2
```

```
x_nega = poly(-xa-c2*(xa**2)-c3*(xa**3),xa,xb)
x_negb = poly(-xb-c2*(xb**2)-c3*(xb**3),xa,xb)
xab = poly(xa+xb+a11*xa*xb+a12*((xa**2)*xb+xa*(xb**2)),xa,xb)
saxb = xab
sbxa = xab
ka = -a11
kb = -a11
```

The function divide_xa (resp. divide_xb) takes as input a polynomial f that is divisible by xa (resp. xb). It then outputs the polynomial g=f/xa (resp. f/xb).

```
def divide_xa(f):
     g = poly(0,xa,xb)
     for i in [1,2,3]:
          for j in [0,1,2,3]:
               C = f.coeff_monomial((xa**i)*(xb**j))
               g = poly(g+C*(xa**(i-1))*(xb**j),xa,xb)
     return g
def divide_xb(f):
     g = poly(0,xa,xb)
     for i in [0,1,2,3]:
          for j in [1,2,3]:
               C = f.coeff_monomial((xa**i)*(xb**j))
               g = poly(g+C*(xa**i)*(xb**(j-1)),xa,xb)
     return g
```

The symbols Da_u and Db_u used in the upcoming code represent truncations of the elements $\Delta_\alpha(u)$ and $\Delta_\beta(u)$, where u lies in the formal group algebra. The matrices Da and Db store the outputs of Da_u and Db_u for all products u of xa and xb up to degree 3. Since x_nega, x_negb, saxb, and sbxa are correct up to and including the degree 3 terms, the outputs of Da_u and Db_u are correct up to and including the degree 2 terms. We use the relation $\Delta_\alpha(uv) = \Delta_\alpha(u)v + s_\alpha(u)\Delta_\alpha(v)$.

```
Da_xa = divide_xa(poly(xa-x_nega,xa,xb))
Da_xb = divide_xa(poly(xb-saxb,xa,xb))
Da_xaxa = Da_xa*xa+x_nega*Da_xa
Da_xaxb = Da_xa*xb+x_nega*Da_xb
Da_xbxb = Da_xb*xb+saxb*Da_xb
Da_xaxaxa = Da_xa*xa*xa+x_nega*Da_xaxa
Da_xaxaxb = Da_xa*xa*xb+x_nega*Da_xaxb
Da_xaxbxb = Da_xa*xb*xb+x_nega*Da_xbxb
Da_xbxbxb = Da_xb*xb*xb+saxb*Da_xbxb
Da = [0,Da_xb,Da_xbxb,Da_xbxbxb,Da_xa,Da_xaxb,Da_xaxbxb,Da_xaxa,Da_xaxaxb,Da_xaxaxa]

Db_xa = divide_xb(poly(xa-sbxa,xa,xb))
Db_xb = divide_xb(poly(xb-x_negb,xa,xb))
Db_xaxa = Db_xa*xa+sbxa*Db_xa
Db_xaxb = Db_xb*xa+x_negb*Db_xa
Db_xbxb = Db_xb*xb+x_negb*Db_xb
Db_xaxaxa = Db_xa*xa*xa+sbxa*Db_xaxa
Db_xaxaxb = Db_xb*xa*xa+x_negb*Db_xaxa
Db_xaxbxb = Db_xb*xa*xb+x_negb*Db_xaxb
Db_xbxbxb = Db_xb*xb*xb+x_negb*Db_xbxb
Db = [0,Db_xb,Db_xbxb,Db_xbxbxb,Db_xa,Db_xaxb,Db_xaxbxb,Db_xaxa,Db_xaxaxb,Db_xaxaxa]
```

The functions compose_Da and compose_Db take as input a polynomial f in xa and xb, and output $\Delta_\alpha(f)$ and $\Delta_\beta(f)$, respectively. Since Da_u and Db_u are correct up to and including the degree 2 terms, and since $\Delta_\alpha(\mathcal{I}_F^i) \subseteq \mathcal{I}_F^{i-1}$ and $\Delta_\beta(\mathcal{I}_F^i) \subseteq \mathcal{I}_F^{i-1}$, the outputs $\Delta_\alpha(f)$ and $\Delta_\beta(f)$ are correct up to and including the degree 2 terms, provided f is correct up to and including degree 3 terms.

```
def compose_Da(f):
     p = 0
     g = poly(0,xa,xb)
```

```
        for i in [0,1,2,3]:
            for j in [0,1,2,3]:
                if ((i+j)<=3):
                    g = poly(g + Da[p]*(f.coeff_monomial((xa**i)*(xb**j))),xa,xb)
                    p = p+1
        return g
    def compose_Db(f):
        p = 0
        g = poly(0,xa,xb)
        for i in [0,1,2,3]:
            for j in [0,1,2,3]:
                if ((i+j)<=3):
                    g = poly(g + Db[p]*(f.coeff_monomial((xa**i)*(xb**j))),xa,xb)
                    p = p+1
        return g
```

The following code computes the constant terms of $\Delta_{I_w}(x_\alpha)$ and $\Delta_{I_w}(x_\beta)$ for reduced sequences I_w over all $w \in W$. It then computes ev_{x_α} and ev_{x_β}.

```
DbDa_xa = compose_Db(Da_xa)
DaDb_xa = compose_Da(Db_xa)
DaDbDa_xa = compose_Da(DbDa_xa)
DbDa_xb = compose_Db(Da_xb)
DaDb_xb = compose_Da(Db_xb)
DaDbDa_xb = compose_Da(DbDa_xb)

da_xa = Da_xa.coeff_monomial(1)
db_xa = Db_xa.coeff_monomial(1)
dadb_xa = DaDb_xa.coeff_monomial(1)
dbda_xa = DbDa_xa.coeff_monomial(1)
dadbda_xa = DaDbDa_xa.coeff_monomial(1)
da_xb = Da_xb.coeff_monomial(1)
db_xb = Db_xb.coeff_monomial(1)
dadb_xb = DaDb_xb.coeff_monomial(1)
dbda_xb = DbDa_xb.coeff_monomial(1)
dadbda_xb = DaDbDa_xb.coeff_monomial(1)

ev_xa = da_xa*Xa+db_xa*Xb+dadb_xa*Xab+dbda_xa*Xba+dadbda_xa*Xaba
ev_xb = da_xb*Xa+db_xb*Xb+dadb_xb*Xab+dbda_xb*Xba+dadbda_xb*Xaba
print("ev_xa = ", ev_xa)
print("ev_xb =", ev_xb)
```

The remaining code computes the $q^v_{w,w'}$ of Equation 5.2.1. The following code computes the constant term of $\Delta_\alpha(s_\beta(x_\alpha))$.

```
Da_sbxa = compose_Da(sbxa)
da_sbxa = Da_sbxa.coeff_monomial(1)
```

The symbol q_w_w'_v represents the coefficient $q^v_{w,w'}$ of Equation 5.2.1. These coefficients can be determined using the output of the code in Appendix B.

```
# XaXb
q_a_a_ab = 0
q_a_b_ab = 1
q_b_b_ab = -da_xb

# XbXa
q_a_a_ba = -db_xa
q_a_b_ba = 1
q_b_b_ba = 0

# XaXbXa
```

```
q_a_a_aba = -dadb_xa-ka*db_xa-ka*db_xa
q_a_b_aba = ka
q_a_ab_aba = 1
q_a_ba_aba = 1 - da_sbxa
q_b_b_aba = 0
q_b_ab_aba = 0
q_b_ba_aba = -da_xb
q_ab_ab_aba = 0
q_ab_ba_aba = 0
q_ba_ba_aba = 0
```

The following code outputs the products $\Delta^*_{I_w} \cdot \Delta^*_{I_{w'}}$.

```
print("XaXa=", q_a_a_ab*Xab+q_a_a_ba*Xba+q_a_a_aba*Xaba)
print("XaXb=", q_a_b_ab*Xab+q_a_b_ba*Xba+q_a_b_aba*Xaba)
print("XaXab=", q_a_ab_aba*Xaba)
print("XaXba=", q_a_ba_aba*Xaba)
print("XaXaba=", 0)
print("XbXb=", q_b_b_ab*Xab+q_b_b_ba*Xba+q_b_b_aba*Xaba)
print("XbXab=", q_b_ab_aba*Xaba)
print("XbXba=", q_b_ba_aba*Xaba)
print("XbXaba=", 0)
print("XabXab=", q_ab_ab_aba*Xaba)
print("XabXba=", q_ab_ba_aba*Xaba)
print("XabXaba=", 0)
print("XbaXba=", q_ba_ba_aba*Xaba)
print("XbaXaba=", 0)
print("XabaXaba=", 0)
```

C.2 $A_1 \times A_1$

```
from sympy import Symbol, poly, div, degree_list
from sympy.abc import x, y

xa = Symbol('x_a')
xb = Symbol('x_b')
Xa = Symbol('X_a')
Xb = Symbol('X_b')
Xab = Symbol('X_ab')
a11 = Symbol('a_11')
c2 = -a11
x_nega = poly(-xa-c2*(xa**2),xa,xb)
x_negb = poly(-xb-c2*(xb**2),xa,xb)

def divide_xa(f):
    g = poly(0,xa,xb)
    for i in [1,2]:
        for j in [0,1,2]:
            C = f.coeff_monomial((xa**i)*(xb**j))
            g = poly(g+C*(xa**(i-1))*(xb**j),xa,xb)
    return g
def divide_xb(f):
    g = poly(0,xa,xb)
    for i in [0,1,2]:
        for j in [1,2]:
            C = f.coeff_monomial((xa**i)*(xb**j))
            g = poly(g+C*(xa**i)*(xb**(j-1)),xa,xb)
    return g

Da_xa = divide_xa(poly(xa-x_nega,xa,xb))
```

```
Da_xb = divide_xa(poly(0,xa,xb))
Da_xaxa = Da_xa*xa+x_nega*Da_xa
Da_xaxb = Da_xa*xb+x_nega*Da_xb
Da_xbxb = Da_xb*xb+xb*Da_xb
Da = [0,Da_xb,Da_xbxb,Da_xa,Da_xaxb,Da_xaxa]

Db_xa = divide_xb(poly(0,xa,xb))
Db_xb = divide_xb(poly(xb-x_negb,xa,xb))
Db_xaxa = Db_xa*xa+xa*Db_xa
Db_xaxb = Db_xb*xa+x_negb*Db_xa
Db_xbxb = Db_xb*xb+x_negb*Db_xb
Db = [0,Db_xb,Db_xbxb,Db_xa,Db_xaxb,Db_xaxa]

def compose_Da(f):
     p = 0
     g = poly(0,xa,xb)
     for i in [0,1,2]:
          for j in [0,1,2]:
               if ((i+j)<=2):
                    g = poly(g + Da[p]*(f.coeff_monomial((xa**i)*(xb**j))),xa,xb)
                    p = p+1
     return g
def compose_Db(f):
     p = 0
     g = poly(0,xa,xb)
     for i in [0,1,2]:
          for j in [0,1,2]:
               if ((i+j)<=2):
                    g = poly(g + Db[p]*(f.coeff_monomial((xa**i)*(xb**j))),xa,xb)
                    p = p+1
     return g

DaDb_xa = compose_Da(Db_xa)
DaDb_xb = compose_Da(Db_xb)

da_xa = Da_xa.coeff_monomial(1)
db_xa = Db_xa.coeff_monomial(1)
dadb_xa = DaDb_xa.coeff_monomial(1)
da_xb = Da_xb.coeff_monomial(1)
db_xb = Db_xb.coeff_monomial(1)
dadb_xb = DaDb_xb.coeff_monomial(1)

ev_xa = da_xa*Xa+db_xa*Xb+dadb_xa*Xab
ev_xb = da_xb*Xa+db_xb*Xb+dadb_xb*Xab
print("ev_xa = ", ev_xa)
print("ev_xb =", ev_xb)

# XaXb
q_a_a_ab = 0
q_a_b_ab = 1
q_b_b_ab = -da_xb

print("XaXa=", q_a_a_ab*Xab)
print("XaXb=", q_a_b_ab*Xab)
print("XaXab=", 0)
print("XbXb=", q_b_b_ab*Xab)
print("XbXab=", 0)
print("XabXab=", 0)
```

C.3 B_2

```
from sympy import Symbol, poly, div, degree_list
from sympy.abc import x, y

xa = Symbol(‘x_a’)
xb = Symbol(‘x_b’)
Xa = Symbol(‘X_a’)
Xb = Symbol(‘X_b’)
Xab = Symbol(‘X_ab’)
Xba = Symbol(‘X_ba’)
Xaba = Symbol(‘X_aba’)
Xbab = Symbol(‘X_bab’)
Xabab = Symbol(‘X_abab’)
a11 = Symbol(‘a_11’)
a12 = Symbol(‘a_12’)
a13 = Symbol(‘a_13’)
a22 = Symbol(‘a_22’)
c2 = -a11
c3 = -a11*c2
c4 = -a11*c3+a12*c2-2*a13+a22
x_nega = poly(-xa-c2*(xa**2)-c3*(xa**3)-c4*(xa**4),xa,xb)
x_negb = poly(-xb-c2*(xb**2)-c3*(xb**3)-c4*(xb**4),xa,xb)
xaa = poly(2*xa+a11*(xa**2)+2*a12*(xa**3)+2*a13*(xa**4)+a22*(xa**4),xa,xb)
xab = poly(xa+xb+a11*xa*xb+a12*((xa**2)*xb+xa*(xb**2))+a13*((xa**3)*xb+xa*(xb**3))
      +a22*(xa**2)*(xb**2),xa,xb)
x2ab = poly(xaa+xb+a11*xaa*xb+a12*((xaa**2)*xb+xaa*(xb**2))+a13*((xaa**3)*xb+xaa*(xb**3))
       +a22*(xaa**2)*(xb**2),xa,xb)
saxb = poly(x2ab,xa,xb)
sbxa = poly(xab,xa,xb)
sasbxa = poly(xab,xa,xb)
sbsaxb = poly(x2ab,xa,xb)
ka = -a11
kb = -a11

def divide_xa(f):
      g = poly(0,xa,xb)
      for i in [1,2,3,4]:
            for j in [0,1,2,3,4]:
                  C = f.coeff_monomial((xa**i)*(xb**j))
                  g = poly(g+C*(xa**(i-1))*(xb**j),xa,xb)
      return g
def divide_xb(f):
      g = poly(0,xa,xb)
      for i in [0,1,2,3,4]:
            for j in [1,2,3,4]:
                  C = f.coeff_monomial((xa**i)*(xb**j))
                  g = poly(g+C*(xa**i)*(xb**(j-1)),xa,xb)
      return g

Da_xa = divide_xa(poly(xa-x_nega,xa,xb))
Da_xb = divide_xa(poly(xb-saxb,xa,xb))
Da_xaxa = Da_xa*xa+x_nega*Da_xa
Da_xaxb = Da_xa*xb+x_nega*Da_xb
Da_xbxb = Da_xb*xb+saxb*Da_xb
Da_xaxaxa = Da_xa*xa*xa+x_nega*Da_xaxa
Da_xaxaxb = Da_xa*xa*xb+x_nega*Da_xaxb
Da_xaxbxb = Da_xa*xb*xb+x_nega*Da_xbxb
Da_xbxbxb = Da_xb*xb*xb+saxb*Da_xbxb
Da_xaxaxaxa = Da_xa*xa*xa*xa+x_nega*Da_xaxaxa
Da_xaxaxaxb = Da_xa*xa*xa*xb+x_nega*Da_xaxaxb
Da_xaxaxbxb = Da_xa*xa*xb*xb+x_nega*Da_xaxbxb
```

```
Da_xaxbxbxb = Da_xa*xb*xb*xb+x_nega*Da_xbxbxb
Da_xbxbxbxb = Da_xb*xb*xb*xb+saxb*Da_xbxbxb
Da = [0,Da_xb,Da_xbxb,Da_xbxbxb,Da_xbxbxbxb,Da_xa,Da_xaxb,Da_xaxbxb,Da_xaxbxbxb,Da_xaxa,
      Da_xaxaxb,Da_xaxaxbxb,Da_xaxaxa,Da_xaxaxaxb,Da_xaxaxaxa]

Db_xa = divide_xb(poly(xa-sbxa,xa,xb))
Db_xb = divide_xb(poly(xb-x_negb,xa,xb))
Db_xaxa = Db_xa*xa+sbxa*Db_xa
Db_xaxb = Db_xb*xa+x_negb*Db_xa
Db_xbxb = Db_xb*xb+x_negb*Db_xb
Db_xaxaxa = Db_xa*xa*xa+sbxa*Db_xaxa
Db_xaxaxb = Db_xb*xa*xa+x_negb*Db_xaxa
Db_xaxbxb = Db_xb*xa*xb+x_negb*Db_xaxb
Db_xbxbxb = Db_xb*xb*xb+x_negb*Db_xbxb
Db_xaxaxaxa = Db_xa*xa*xa*xa+sbxa*Db_xaxaxa
Db_xaxaxaxb = Db_xb*xa*xa*xa+x_negb*Db_xaxaxa
Db_xaxaxbxb = Db_xb*xa*xa*xb+x_negb*Db_xaxaxb
Db_xaxbxbxb = Db_xb*xa*xb*xb+x_negb*Db_xaxbxb
Db_xbxbxbxb = Db_xb*xb*xb*xb+x_negb*Db_xbxbxb
Db = [0,Db_xb,Db_xbxb,Db_xbxbxb,Db_xbxbxbxb,Db_xa,Db_xaxb,Db_xaxbxb,Db_xaxbxbxb,
      Db_xaxa,Db_xaxaxb,Db_xaxaxbxb,Db_xaxaxa,Db_xaxaxaxb,Db_xaxaxaxa]

def compose_Da(f):
     p = 0
     g = poly(0,xa,xb)
     for i in [0,1,2,3,4]:
          for j in [0,1,2,3,4]:
               if ((i+j)<=4):
                    g = poly(g + Da[p]*(f.coeff_monomial((xa**i)*(xb**j))),xa,xb)
                    p = p+1
     return g
def compose_Db(f):
     p = 0
     g = poly(0,xa,xb)
     for i in [0,1,2,3,4]:
          for j in [0,1,2,3,4]:
               if ((i+j)<=4):
                    g = poly(g + Db[p]*(f.coeff_monomial((xa**i)*(xb**j))),xa,xb)
                    p = p+1
     return g

DbDa_xa = compose_Db(Da_xa)
DaDb_xa = compose_Da(Db_xa)
DbDaDb_xa = compose_Db(DaDb_xa)
DaDbDa_xa = compose_Da(DbDa_xa)
DaDbDaDb_xa = compose_Da(DbDaDb_xa)
DbDa_xb = compose_Db(Da_xb)
DaDb_xb = compose_Da(Db_xb)
DbDaDb_xb = compose_Db(DaDb_xb)
DaDbDa_xb = compose_Da(DbDa_xb)
DaDbDaDb_xb = compose_Da(DbDaDb_xb)

da_xa = Da_xa.coeff_monomial(1)
db_xa = Db_xa.coeff_monomial(1)
dadb_xa = DaDb_xa.coeff_monomial(1)
dbda_xa = DbDa_xa.coeff_monomial(1)
dadbda_xa = DaDbDa_xa.coeff_monomial(1)
dbdadb_xa = DbDaDb_xa.coeff_monomial(1)
dadbdadb_xa = DaDbDaDb_xa.coeff_monomial(1)
da_xb = Da_xb.coeff_monomial(1)
db_xb = Db_xb.coeff_monomial(1)
dadb_xb = DaDb_xb.coeff_monomial(1)
```

```
dbda_xb = DbDa_xb.coeff_monomial(1)
dadbda_xb = DaDbDa_xb.coeff_monomial(1)
dbdadb_xb = DbDaDb_xb.coeff_monomial(1)
dadbdadb_xb = DaDbDaDb_xb.coeff_monomial(1)

ev_xa = da_xa*Xa+db_xa*Xb+dadb_xa*Xab+dbda_xa*Xba+dadbda_xa*Xaba+dbdadb_xa*Xbab+dadbdadb_xa*Xabab
ev_xb = da_xb*Xa+db_xb*Xb+dadb_xb*Xab+dbda_xb*Xba+dadbda_xb*Xaba+dbdadb_xb*Xbab+dadbdadb_xb*Xabab
print("ev_xa = ", ev_xa)
print("ev_xb =", ev_xb)

Da_sbxa = compose_Da(sbxa)
Db_saxb = compose_Db(saxb)
Da_sbsaxb = compose_Da(sbsaxb)
da_sbxa = Da_sbxa.coeff_monomial(1)
db_saxb = Db_saxb.coeff_monomial(1)
da_sbsaxb = Da_sbsaxb.coeff_monomial(1)

# XaXb
q_a_a_ab = 0
q_a_b_ab = 1
q_b_b_ab = -da_xb

# XbXa
q_a_a_ba = -db_xa
q_a_b_ba = 1
q_b_b_ba = 0

# XaXbXa
q_a_a_aba = -dadb_xa-ka*db_xa-ka*db_xa
q_a_b_aba = ka
q_a_ab_aba = 1
q_a_ba_aba = 1 - da_sbxa
q_b_b_aba = 0
q_b_ab_aba = 0
q_b_ba_aba = -da_xb
q_ab_ab_aba = 0
q_ab_ba_aba = 0
q_ba_ba_aba = 0

# XbXaXb
q_a_a_bab = 0
q_a_b_bab = kb
q_a_ab_bab = -db_xa
q_a_ba_bab = 0
q_b_b_bab = -dbda_xb-kb*da_xb-kb*da_xb
q_b_ab_bab = 1-db_saxb
q_b_ba_bab = 1
q_ab_ab_bab = 0
q_ab_ba_bab = 0
q_ba_ba_bab = 0

# XaXbXaXb
q_a_a_abab = Symbol('q_a_a_abab')
q_a_b_abab = Symbol('q_a_b_abab')
q_a_ab_abab = Symbol('q_a_ab_abab')
q_a_ba_abab = Symbol('q_a_ba_abab')
q_a_aba_abab = 0
q_a_bab_abab = 1-da_sbxa
q_b_b_abab = Symbol('q_b_b_abab')
q_b_ab_abab = Symbol('q_b_ab_abab')
q_b_ba_abab = Symbol('q_b_ba_abab')
q_b_aba_abab = 1
```

```
q_b_bab_abab = -da_xb-da_sbsaxb
q_ab_ab_abab = Symbol('q_ab_ab_abab')
q_ab_ba_abab = Symbol('q_ab_ba_abab')
q_ab_aba_abab = 0
q_ab_bab_abab = 0
q_ba_ba_abab = Symbol('q_ba_ba_abab')
q_ba_aba_abab = 0
q_ba_bab_abab = 0
q_aba_aba_abab = 0
q_aba_bab_abab = 0
q_bab_bab_abab = 0

print("XaXa=", q_a_a_ab*Xab+ q_a_a_ba*Xba+q_a_a_aba*Xaba+q_a_a_bab*Xbab+q_a_a_abab*Xabab)
print("XaXb=", q_a_b_ab*Xab+q_a_b_ba*Xba+q_a_b_aba*Xaba+q_a_b_bab*Xbab+q_a_b_abab*Xabab)
print("XaXab=", q_a_ab_aba*Xaba+q_a_ab_bab*Xbab+q_a_ab_abab*Xabab)
print("XaXba=", q_a_ba_aba*Xaba+q_a_ba_bab*Xbab+q_a_ba_abab*Xabab)
print("XaXaba=", q_a_aba_abab*Xabab)
print("XaXbab=", q_a_bab_abab*Xabab)
print("XaXabab=", 0)
print("XbXb=", q_b_b_ab*Xab+q_b_b_ba*Xba+q_b_b_aba*Xaba+q_b_b_bab*Xbab+q_b_b_abab*Xabab)
print("XbXab=", q_b_ab_aba*Xaba+q_b_ab_bab*Xbab+q_b_ab_abab*Xabab)
print("XbXba=", q_b_ba_aba*Xaba+q_b_ba_bab*Xbab+q_b_ba_abab*Xabab)
print("XbXaba=", q_b_aba_abab*Xabab)
print("XbXbab=", q_b_bab_abab*Xabab)
print("XbXabab=", 0)
print("XabXab=", q_ab_ab_aba*Xaba+q_ab_ab_bab*Xbab+q_ab_ab_abab*Xabab)
print("XabXba=", q_ab_ba_aba*Xaba+q_ab_ba_bab*Xbab+q_ab_ba_abab*Xabab)
print("XabXaba=", q_ab_aba_abab*Xabab)
print("XabXbab=", q_ab_bab_abab*Xabab)
print("XabXabab=", 0)
print("XbaXba=", q_ba_ba_aba*Xaba+q_ba_ba_bab*Xbab+q_ba_ba_abab*Xabab)
print("XbaXaba=", q_ba_aba_abab*Xabab)
print("XbaXbab=", q_ba_bab_abab*Xabab)
print("XbaXabab=", 0)
print("XabaXaba=", q_aba_aba_abab*Xabab)
print("XabaXbab=", q_aba_bab_abab*Xabab)
print("XabaXabab=", 0)
print("XbabXbab=", q_bab_bab_abab*Xabab)
print("XbabXabab=", 0)
print("XababXabab=", 0)
```

C.4 G_2

```
from sympy import Symbol, poly, div, degree_list
from sympy.abc import x, y

xa = Symbol('x_a')
xb = Symbol('x_b')
Xa = Symbol('X_a')
Xb = Symbol('X_b')
Xab = Symbol('X_ab')
Xba = Symbol('X_ba')
Xaba = Symbol('X_aba')
Xbab = Symbol('X_bab')
Xabab = Symbol('X_abab')
Xbaba = Symbol('X_baba')
Xababa = Symbol('X_ababa')
Xbabab = Symbol('X_babab')
Xababab = Symbol('X_ababab')
```

```
a11 = Symbol('a_11')
a12 = Symbol('a_12')
a13 = Symbol('a_13')
a22 = Symbol('a_22')
a14 = Symbol('a_14')
a23 = Symbol('a_23')
a24 = Symbol('a_24')
a33 = Symbol('a_33')
c2 = -a11
c3 = -a11*c2
c4 = -a11*c3+a12*c2-2*a13+a22
c5 = -a11*c4+a12*c3+a12*c2**2-4*a13*c2+2*a22*c2
c6 = -a11*c5+a12*c4+2*a12*c2*c3-4*a13*c3-3*a13*c2**2+3*a14*c2-2*a15+2*a22*c3+
     a22*c2**2-a23*c2+2*a24-a33
```

The function truncate() takes as input a polynomial f in xa and xb, and outputs the truncation of f to degree 6.

```
def truncate(f):
g = poly(0,xa,xb)
for i in [0,1,2,3,4,5,6]:
     for j in [0,1,2,3,4,5,6]:
          if ((i+j) <= 6):
               C = f.coeff_monomial((xa**i)*(xb**j))
               g = poly(g+C*(xa**i)*(xb**j),xa,xb)
return g

x_nega = poly(-xa-c2*(xa**2)-c3*(xa**3)-c4*(xa**4)-c5*(xa**5)-c6*(xa**6),xa,xb)
x_negb = poly(-xb-c2*(xb**2)-c3*(xb**3)-c4*(xb**4)-c5*(xb**5)-c6*(xb**6),xa,xb)
xbb = poly(2*xb+a11*(xb**2)+2*a12*(xb**3)+2*a13*(xb**4)+a22*(xb**4)+2*a14*(xb**5)
     +2*a23*(xb**5)+2*a15*(xb**6)+2*a24*(xb**6)+a33*(xb**6),xa,xb)
xaa = poly(2*xa+a11*(xa**2)+2*a12*(xa**3)+2*a13*(xa**4)+a22*(xa**4)+2*a14*(xa**5)+
     2*a23*(xa**5)+2*a15*(xa**6)+2*a24*(xa**6)+a33*(xa**6),xa,xb)
xab = poly(xa+xb+a11*xa*xb+a12*((xa**2)*xb+xa*(xb**2))+a13*((xa**3)*xb+xa*(xb**3))+
     a22*(xa**2)*(xb**2)+a14*((xa**4)*xb+xa*(xb**4))+a23*((xa**2)*(xb**3)+(xa**3)*(xb**2))+
     a15*((xa**5)*xb+xa*(xb**5))+a24*((xa**2)*(xb**4)+(xa**4)*(xb**2))+a33*(xa**3)*(xb**3),xa,xb)
xa3b_long = poly(xbb+xab+a11*xbb*xab+a12*((xbb**2)*xab+xbb*(xab**2))+a13*((xbb**3)*xab+
          xbb*(xab**3))+a22*(xbb**2)*(xab**2)+a14*((xab)*(xbb**4)+(xab**4)*xbb)+
          a23*((xab**2)*(xbb**3)+(xab**3)*(xbb**2))+a15*((xbb**5)*xab+xbb*(xab**5))+
          a24*((xbb**4)*(xab**2)+(xbb**2)*(xab**4))+a33*(xbb**3)*(xab**3),xa,xb)
xa3b = truncate(xa3b_long)
xa2b_long = poly(xbb+xa+a11*xbb*xa+a12*((xbb**2)*xa+xbb*(xa**2))+a13*((xbb**3)*xa+xbb*(xa**3))+
          a22*(xbb**2)*(xa**2)+a14*((xa)*(xbb**4)+(xa**4)*xbb)+a23*((xa**2)*(xbb**3)+
          (xa**3)*(xbb**2))+a15*((xbb**5)*xa+xbb*(xa**5))+a24*((xbb**4)*(xa**2)+
          (xbb**2)*(xa**4))+a33*(xbb**3)*(xa**3),xa,xb)
xa2b = truncate(xa2b_long)
x2a3b_long = poly(xa+xa3b+a11*xa*xa3b+a12*((xa**2)*xa3b+xa*(xa3b**2))+a13*((xa**3)*xa3b+
           xa*(xa3b**3))+a22*(xa**2)*(xa3b**2)+a14*(xa*(xa3b**4)+(xa**4)*xa3b)+
           a23*((xa**2)*(xa3b**3)+(xa**3)*(xa3b**2))+a15*((xa**5)*xa3b+xa*(xa3b**5))+
           a24*((xa**4)*(xa3b**2)+(xa**2)*(xa3b**4))+a33*(xa**3)*(xa3b**3),xa,xb)
x2a3b = truncate(x2a3b_long)
saxb = xab
sbsaxb = xa2b
sasbsaxb = xa2b
sbsasbsaxb = xab
sbxa = xa3b
sasbxa = x2a3b
sbsasbxa = x2a3b
sasbsasbxa = xa3b
ka = -a11
kb = -a11

def divide_xa(f):
```

```
     g = poly(0,xa,xb)
     for i in [1,2,3,4,5,6]:
          for j in [0,1,2,3,4,5,6]:
               C = f.coeff_monomial((xa**i)*(xb**j))
               g = poly(g+C*(xa**(i-1))*(xb**j),xa,xb)
     return g
def divide_xb(f):
     g = poly(0,xa,xb)
     for i in [0,1,2,3,4,5,6]:
          for j in [1,2,3,4,5,6]:
               C = f.coeff_monomial((xa**i)*(xb**j))
               g = poly(g+C*(xa**i)*(xb**(j-1)),xa,xb)
     return g

Da_xa = truncate(divide_xa(poly(xa-x_nega,xa,xb)))
Da_xb = truncate(divide_xa(poly(xb-saxb,xa,xb)))
Da_xaxa = truncate(Da_xa*xa+x_nega*Da_xa)
Da_xaxb = truncate(Da_xa*xb+x_nega*Da_xb)
Da_xbxb = truncate(Da_xb*xb+saxb*Da_xb)
Da_xaxaxa = truncate(Da_xa*xa*xa+x_nega*Da_xaxa)
Da_xaxaxb = truncate(Da_xa*xa*xb+x_nega*Da_xaxb)
Da_xaxbxb = truncate(Da_xa*xb*xb+x_nega*Da_xbxb)
Da_xbxbxb = truncate(Da_xb*xb*xb+saxb*Da_xbxb)
Da_xaxaxaxa = truncate(Da_xa*xa*xa*xa+x_nega*Da_xaxaxa)
Da_xaxaxaxb = truncate(Da_xa*xa*xa*xb+x_nega*Da_xaxaxb)
Da_xaxaxbxb = truncate(Da_xa*xa*xb*xb+x_nega*Da_xaxbxb)
Da_xaxbxbxb = truncate(Da_xa*xb*xb*xb+x_nega*Da_xbxbxb)
Da_xbxbxbxb = truncate(Da_xb*xb*xb*xb+saxb*Da_xbxbxb)
Da_xaxaxaxaxa = truncate(Da_xa*xa*xa*xa*xa+x_nega*Da_xaxaxaxa)
Da_xaxaxaxaxb = truncate(Da_xa*xa*xa*xa*xb+x_nega*Da_xaxaxaxb)
Da_xaxaxaxbxb = truncate(Da_xa*xa*xa*xb*xb+x_nega*Da_xaxaxbxb)
Da_xaxaxbxbxb = truncate(Da_xa*xa*xb*xb*xb+x_nega*Da_xaxbxbxb)
Da_xaxbxbxbxb = truncate(Da_xa*xb*xb*xb*xb+x_nega*Da_xbxbxbxb)
Da_xbxbxbxbxb = truncate(Da_xb*xb*xb*xb*xb+saxb*Da_xbxbxbxb)
Da_xaxaxaxaxaxa = truncate(Da_xa*xa*xa*xa*xa*xa+x_nega*Da_xaxaxaxaxa)
Da_xaxaxaxaxaxb = truncate(Da_xa*xa*xa*xa*xa*xb+x_nega*Da_xaxaxaxaxb)
Da_xaxaxaxaxbxb = truncate(Da_xa*xa*xa*xa*xb*xb+x_nega*Da_xaxaxaxbxb)
Da_xaxaxaxbxbxb = truncate(Da_xa*xa*xa*xb*xb*xb+x_nega*Da_xaxaxbxbxb)
Da_xaxaxbxbxbxb = truncate(Da_xa*xa*xb*xb*xb*xb+x_nega*Da_xaxbxbxbxb)
Da_xaxbxbxbxbxb = truncate(Da_xa*xb*xb*xb*xb*xb+x_nega*Da_xbxbxbxbxb)
Da_xbxbxbxbxbxb = truncate(Da_xb*xb*xb*xb*xb*xb+saxb*Da_xbxbxbxbxb)
Da = [0,Da_xb,Da_xbxb,Da_xbxbxb,Da_xbxbxbxb,Da_xbxbxbxbxb,Da_xbxbxbxbxbxb,Da_xa,Da_xaxb,Da_xaxbxb,
     Da_xaxbxbxb,Da_xaxbxbxbxb,Da_xaxbxbxbxbxb,Da_xaxa,Da_xaxaxb,Da_xaxaxbxb,Da_xaxaxbxbxb,
     Da_xaxaxbxbxbxb,Da_xaxaxa,Da_xaxaxaxb,Da_xaxaxaxbxb,Da_xaxaxaxbxbxb,Da_xaxaxaxa,Da_xaxaxaxaxb,
     Da_xaxaxaxaxbxb,Da_xaxaxaxaxa,Da_xaxaxaxaxaxb,Da_xaxaxaxaxaxa]

Db_xa = truncate(divide_xb(poly(xa-sbxa,xa,xb)))
Db_xb = truncate(divide_xb(poly(xb-x_negb,xa,xb)))
Db_xaxa = truncate(Db_xa*xa+sbxa*Db_xa)
Db_xaxb = truncate(Db_xb*xa+x_negb*Db_xa)
Db_xbxb = truncate(Db_xb*xb+x_negb*Db_xb)
Db_xaxaxa = truncate(Db_xa*xa*xa+sbxa*Db_xaxa)
Db_xaxaxb = truncate(Db_xb*xa*xa+x_negb*Db_xaxa)
Db_xaxbxb = truncate(Db_xb*xa*xb+x_negb*Db_xaxb)
Db_xbxbxb = truncate(Db_xb*xb*xb+x_negb*Db_xbxb)
Db_xaxaxaxa = truncate(Db_xa*xa*xa*xa+sbxa*Db_xaxaxa)
Db_xaxaxaxb = truncate(Db_xb*xa*xa*xa+x_negb*Db_xaxaxa)
Db_xaxaxbxb = truncate(Db_xb*xa*xa*xb+x_negb*Db_xaxaxb)
Db_xaxbxbxb = truncate(Db_xb*xa*xb*xb+x_negb*Db_xaxbxb)
Db_xbxbxbxb = truncate(Db_xb*xb*xb*xb+x_negb*Db_xbxbxb)
Db_xaxaxaxaxa = truncate(Db_xa*xa*xa*xa*xa+sbxa*Db_xaxaxaxa)
Db_xaxaxaxaxb = truncate(Db_xb*xa*xa*xa*xa+x_negb*Db_xaxaxaxa)
```

```
Db_xaxaxaxbxb = truncate(Db_xb*xa*xa*xa*xb+x_negb*Db_xaxaxaxb)
Db_xaxaxbxbxb = truncate(Db_xb*xa*xa*xb*xb+x_negb*Db_xaxaxbxb)
Db_xaxbxbxbxb = truncate(Db_xb*xa*xb*xb*xb+x_negb*Db_xaxbxbxb)
Db_xbxbxbxbxb = truncate(Db_xb*xb*xb*xb*xb+x_negb*Db_xbxbxbxb)
Db_xaxaxaxaxaxa = truncate(Db_xa*xa*xa*xa*xa*xa+sbxa*Db_xaxaxaxaxa)
Db_xaxaxaxaxaxb = truncate(Db_xb*xa*xa*xa*xa*xa+x_negb*Db_xaxaxaxaxa)
Db_xaxaxaxaxbxb = truncate(Db_xb*xa*xa*xa*xa*xb+x_negb*Db_xaxaxaxaxb)
Db_xaxaxaxbxbxb = truncate(Db_xb*xa*xa*xa*xb*xb+x_negb*Db_xaxaxaxbxb)
Db_xaxaxbxbxbxb = truncate(Db_xb*xa*xa*xb*xb*xb+x_negb*Db_xaxaxbxbxb)
Db_xaxbxbxbxbxb = truncate(Db_xb*xa*xb*xb*xb*xb+x_negb*Db_xaxbxbxbxb)
Db_xbxbxbxbxbxb = truncate(Db_xb*xb*xb*xb*xb*xb+x_negb*Db_xbxbxbxbxb)
Db = [0,Db_xb,Db_xbxb,Db_xbxbxb,Db_xbxbxbxb,Db_xbxbxbxbxb,Db_xbxbxbxbxbxb,Db_xa,Db_xaxb,Db_xaxbxb,
    Db_xaxbxbxb,Db_xaxbxbxbxb,Db_xaxbxbxbxbxb,Db_xaxa,Db_xaxaxb,Db_xaxaxbxb,Db_xaxaxbxbxb,
    Db_xaxaxbxbxbxb,Db_xaxaxa,Db_xaxaxaxb,Db_xaxaxaxbxb,Db_xaxaxaxbxbxb,Db_xaxaxaxa,Db_xaxaxaxaxb,
    Db_xaxaxaxaxbxb,Db_xaxaxaxaxa,Db_xaxaxaxaxaxb,Db_xaxaxaxaxaxa]

def compose_Da(f):
    p = 0
    g = poly(0,xa,xb)
    for i in [0,1,2,3,4,5,6]:
        for j in [0,1,2,3,4,5,6]:
            if ((i+j)<=6):
                g = poly(g + Da[p]*(f.coeff_monomial((xa**i)*(xb**j))),xa,xb)
                p = p+1
    return g
def compose_Db(f):
    p = 0
    g = poly(0,xa,xb)
    for i in [0,1,2,3,4,5,6]:
        for j in [0,1,2,3,4,5,6]:
            if ((i+j)<=6):
                g = poly(g + Db[p]*(f.coeff_monomial((xa**i)*(xb**j))),xa,xb)
                p = p+1
    return g

DbDa_xa = compose_Db(Da_xa)
DaDb_xa = compose_Da(Db_xa)
DbDaDb_xa = compose_Db(DaDb_xa)
DaDbDa_xa = compose_Da(DbDa_xa)
DaDbDaDb_xa = compose_Da(DbDaDb_xa)
DbDaDbDa_xa = compose_Db(DaDbDa_xa)
DaDbDaDbDa_xa = compose_Da(DbDaDbDa_xa)
DbDaDbDaDb_xa = compose_Db(DaDbDaDb_xa)
DaDbDaDbDaDb_xa = compose_Da(DbDaDbDaDb_xa)
DbDa_xb = compose_Db(Da_xb)
DaDb_xb = compose_Da(Db_xb)
DbDaDb_xb = compose_Db(DaDb_xb)
DaDbDa_xb = compose_Da(DbDa_xb)
DaDbDaDb_xb = compose_Da(DbDaDb_xb)
DbDaDbDa_xb = compose_Db(DaDbDa_xb)
DaDbDaDbDa_xb = compose_Da(DbDaDbDa_xb)
DbDaDbDaDb_xb = compose_Db(DaDbDaDb_xb)
DaDbDaDbDaDb_xb = compose_Da(DbDaDbDaDb_xb)

da_xa = Da_xa.coeff_monomial(1)
db_xa = Db_xa.coeff_monomial(1)
dadb_xa = DaDb_xa.coeff_monomial(1)
dbda_xa = DbDa_xa.coeff_monomial(1)
dadbda_xa = DaDbDa_xa.coeff_monomial(1)
dbdadb_xa = DbDaDb_xa.coeff_monomial(1)
dadbdadb_xa = DaDbDaDb_xa.coeff_monomial(1)
dbdadbda_xa = DbDaDbDa_xa.coeff_monomial(1)
```

```
dadbdadbda_xa = DaDbDaDbDa_xa.coeff_monomial(1)
dbdadbdadb_xa = DbDaDbDaDb_xa.coeff_monomial(1)
dadbdadbdadb_xa = DaDbDaDbDaDb_xa.coeff_monomial(1)
da_xb = Da_xb.coeff_monomial(1)
db_xb = Db_xb.coeff_monomial(1)
dadb_xb = DaDb_xb.coeff_monomial(1)
dbda_xb = DbDa_xb.coeff_monomial(1)
dadbda_xb = DaDbDa_xb.coeff_monomial(1)
dbdadb_xb = DbDaDb_xb.coeff_monomial(1)
dadbdadb_xb = DaDbDaDb_xb.coeff_monomial(1)
dbdadbda_xb = DbDaDbDa_xb.coeff_monomial(1)
dadbdadbda_xb = DaDbDaDbDa_xb.coeff_monomial(1)
dbdadbdadb_xb = DbDaDbDaDb_xb.coeff_monomial(1)
dadbdadbdadb_xb = DaDbDaDbDaDb_xb.coeff_monomial(1)

ev_xa = da_xa*Xa+db_xa*Xb+dadb_xa*Xab+dbda_xa*Xba+dadbda_xa*Xaba+dbdadb_xa*Xbab+dadbdadb_xa*Xabab+
        dbdadbda_xa*Xbaba+dadbdadbda_xa*Xababa+dbdadbdadb_xa*Xbabab+dadbdadbdadb_xa*Xababab
ev_xb = da_xb*Xa+db_xb*Xb+dadb_xb*Xab+dbda_xb*Xba+dadbda_xb*Xaba+dbdadb_xb*Xbab+dadbdadb_xb*Xabab+
        dbdadbda_xb*Xbaba+dadbdadbda_xb*Xababa+dbdadbdadb_xb*Xbabab+dadbdadbdadb_xb*Xababab
print("ev_xa = ", ev_xa)
print("ev_xb = ", ev_xb)

Da_sbxa = compose_Da(sbxa)
Db_saxb = compose_Db(saxb)
Db_sasbxa = compose_Db(sasbxa)
Da_sbsaxb = compose_Da(sbsaxb)
Da_sbsasbxa = compose_Da(sbsasbxa)
Da_sbsasbsaxb = compose_Da(sbsasbsaxb)
Db_sasbsaxb = compose_Db(sasbsaxb)
DaDb_saxb = compose_Da(compose_Db(saxb))
DaDb_xasaxb = compose_Da(compose_Db(xa*saxb))
DbDa_xbsaxb = compose_Db(compose_Da(xb*saxb))
DbDa_xbsbxa = compose_Db(compose_Da(xb*sbxa))
DbDa_sbxa = compose_Db(compose_Da(sbxa))

db_saxb = Db_saxb.coeff_monomial(1)
da_sbxa = Da_sbxa.coeff_monomial(1)
db_sasbxa = Db_sasbxa.coeff_monomial(1)
da_sbsaxb = Da_sbsaxb.coeff_monomial(1)
da_sbsasbxa = Da_sbsasbxa.coeff_monomial(1)
da_sbsasbsaxb = Da_sbsasbsaxb.coeff_monomial(1)
db_sasbsaxb = Db_sasbsaxb.coeff_monomial(1)
dadb_saxb = DaDb_saxb.coeff_monomial(1)
dadb_xasaxb = DaDb_xasaxb.coeff_monomial(1)
dbda_xbsaxb = DbDa_xbsaxb.coeff_monomial(1)
dbda_xbsbxa = DbDa_xbsbxa.coeff_monomial(1)
dbda_sbxa = DbDa_sbxa.coeff_monomial(1)

# XaXb
q_a_a_ab = 0
q_a_b_ab = 1
q_b_b_ab = -da_xb

# XbXa
q_a_a_ba = -db_xa
q_a_b_ba = 1
q_b_b_ba = 0

# XaXbXa
q_a_a_aba = -dadb_xa-ka*db_xa-ka*db_xa
q_a_b_aba = ka
q_a_ab_aba = 1
```

```
q_a_ba_aba = 1 - da_sbxa
q_b_b_aba = 0
q_b_ab_aba = 0
q_b_ba_aba = -da_xb
q_ab_ab_aba = 0
q_ab_ba_aba = 0
q_ba_ba_aba = 0

# XbXaXb
q_a_a_bab = 0
q_a_b_bab = kb
q_a_ab_bab = -db_xa
q_a_ba_bab = 0
q_b_b_bab = -dbda_xb-kb*da_xb-kb*da_xb
q_b_ab_bab = 1-db_saxb
q_b_ba_bab = 1
q_ab_ab_bab = 0
q_ab_ba_bab = 0
q_ba_ba_bab = 0

# XaXbXaXb
q_a_a_abab = Symbol('q_a_a_abab')
q_a_b_abab = Symbol('q_a_b_abab')
q_a_ab_abab = Symbol('q_a_ab_abab')
q_a_ba_abab = Symbol('q_a_ba_abab')
q_a_aba_abab = 0
q_a_bab_abab = 1-da_sbxa
q_b_b_abab = Symbol('q_b_b_abab')
q_b_ab_abab = Symbol('q_b_ab_abab')
q_b_ba_abab = Symbol('q_b_ba_abab')
q_b_aba_abab = 1
q_b_bab_abab = -da_xb-da_sbsaxb
q_ab_ab_abab = Symbol('q_ab_ab_abab')
q_ab_ba_abab = Symbol('q_ab_ba_abab')
q_ab_aba_abab = 0
q_ab_bab_abab = 0
q_ba_ba_abab = Symbol('q_ba_ba_abab')
q_ba_aba_abab = 0
q_ba_bab_abab = 0
q_aba_aba_abab = 0
q_aba_bab_abab = 0
q_bab_bab_abab = 0

# XbXaXbXa
q_a_a_baba = Symbol('q_a_a_baba')
q_a_b_baba = Symbol('q_a_b_baba')
q_a_ab_baba = Symbol('q_a_ab_baba')
q_a_ba_baba = Symbol('q_a_ba_baba')
q_a_aba_baba = -db_xa-db_sasbxa
q_a_bab_baba = 1
q_b_b_baba = Symbol('q_b_b_baba')
q_b_ab_baba = Symbol('q_b_ab_baba')
q_b_ba_baba = Symbol('q_b_ba_baba')
q_b_aba_baba = 1-db_saxb
q_b_bab_baba = 0
q_ab_ab_baba = Symbol('q_ab_ab_baba')
q_ab_ba_baba = Symbol('q_ab_ba_baba')
q_ab_aba_baba = 0
q_ab_bab_baba = 0
q_ba_ba_baba = Symbol('q_ba_ba_baba')
q_ba_aba_baba = 0
q_ba_bab_baba = 0
```

```
q_aba_aba_baba = 0
q_aba_bab_baba = 0
q_bab_bab_baba = 0

# XaXbXaXbXa
q_a_a_ababa = Symbol('q_a_a_ababa')
q_a_b_ababa = Symbol('q_a_b_ababa')
q_a_ab_ababa = Symbol('q_a_ab_ababa')
q_a_ba_ababa = Symbol('q_a_ba_ababa')
q_a_aba_ababa = Symbol('q_a_aba_ababa')
q_a_bab_ababa = Symbol('q_a_bab_ababa')
q_a_abab_ababa = 1
q_a_baba_ababa = 1-da_sbxa-da_sbsasbxa
q_b_b_ababa = Symbol('q_b_b_ababa')
q_b_ab_ababa = Symbol('q_b_ab_ababa')
q_b_ba_ababa = Symbol('q_b_ba_ababa')
q_b_aba_ababa = Symbol('q_b_aba_ababa')
q_b_bab_ababa = Symbol('q_b_bab_ababa')
q_b_abab_ababa = 0
q_b_baba_ababa = -da_xb-da_sbsaxb
q_ab_ab_ababa = Symbol('q_ab_ab_ababa')
q_ab_ba_ababa = Symbol('q_ab_ba_ababa')
q_ab_aba_ababa = Symbol('q_ab_aba_ababa')
q_ab_bab_ababa = Symbol('q_ab_bab_ababa')
q_ab_abab_ababa = 0
q_ab_baba_ababa = 0
q_ba_ba_ababa = Symbol('q_ba_ba_ababa')
q_ba_aba_ababa = Symbol('q_ba_aba_ababa')
q_ba_bab_ababa = Symbol('q_ba_bab_ababa')
q_ba_abab_ababa = 0
q_ba_baba_ababa = 0
q_aba_aba_ababa = Symbol('q_aba_aba_ababa')
q_aba_bab_ababa = Symbol('q_aba_bab_ababa')
q_aba_abab_ababa = 0
q_aba_baba_ababa = 0
q_bab_bab_ababa = Symbol('q_bab_bab_ababa')
q_bab_abab_ababa = 0
q_bab_baba_ababa = 0
q_abab_abab_ababa = 0
q_abab_baba_ababa = 0
q_baba_baba_ababa = 0

# XbXaXbXaXb
q_a_a_babab = Symbol('q_a_a_babab')
q_a_b_babab = Symbol('q_a_b_babab')
q_a_ab_babab = Symbol('q_a_ab_babab')
q_a_ba_babab = Symbol('q_a_ba_babab')
q_a_aba_babab = Symbol('q_a_aba_babab')
q_a_bab_babab = Symbol('q_a_bab_babab')
q_a_abab_babab = -db_xa-db_sasbxa
q_a_baba_babab = 0
q_b_b_babab = Symbol('q_b_b_babab')
q_b_ab_babab = Symbol('q_b_ab_babab')
q_b_ba_babab = Symbol('q_b_ba_babab')
q_b_aba_babab = Symbol('q_b_aba_babab')
q_b_bab_babab = Symbol('q_b_bab_babab')
q_b_abab_babab = 1-db_saxb-db_sasbsaxb
q_b_baba_babab = 1
q_ab_ab_babab = Symbol('q_ab_ab_babab')
q_ab_ba_babab = Symbol('q_ab_ba_babab')
q_ab_aba_babab = Symbol('q_ab_aba_babab')
q_ab_bab_babab = Symbol('q_ab_bab_babab')
```

```
q_ab_abab_babab = 0
q_ab_baba_babab = 0
q_ba_ba_babab = Symbol('q_ba_ba_babab')
q_ba_aba_babab = Symbol('q_ba_aba_babab')
q_ba_bab_babab = Symbol('q_ba_bab_babab')
q_ba_abab_babab = 0
q_ba_baba_babab = 0
q_aba_aba_babab = Symbol('q_aba_aba_babab')
q_aba_bab_babab = Symbol('q_aba_bab_babab')
q_aba_abab_babab = 0
q_aba_baba_babab = 0
q_bab_bab_babab = Symbol('q_bab_bab_babab')
q_bab_abab_babab = 0
q_bab_baba_babab = 0
q_abab_abab_babab = 0
q_abab_baba_babab = 0
q_baba_baba_babab = 0

# XaXbXaXbXaXb
q_a_a_ababab = Symbol('q_a_a_ababab')
q_a_b_ababab = Symbol('q_a_b_ababab')
q_a_ab_ababab = Symbol('q_a_ab_ababab')
q_a_ba_ababab = Symbol('q_a_ba_ababab')
q_a_aba_ababab = Symbol('q_a_aba_ababab')
q_a_bab_ababab = Symbol('q_a_bab_ababab')
q_a_abab_ababab = Symbol('q_a_abab_ababab')
q_a_baba_ababab = Symbol('q_a_baba_ababab')
q_a_ababa_ababab = 0
q_a_babab_ababab = 1-da_sbxa-da_sbsasbxa
q_b_b_ababab = Symbol('q_b_b_ababab')
q_b_ab_ababab = Symbol('q_b_ab_ababab')
q_b_ba_ababab = Symbol('q_b_ba_ababab')
q_b_aba_ababab = Symbol('q_b_aba_ababab')
q_b_bab_ababab = Symbol('q_b_bab_ababab')
q_b_abab_ababab = Symbol('q_b_abab_ababab')
q_b_baba_ababab = Symbol('q_b_baba_ababab')
q_b_ababa_ababab = 1
q_b_babab_ababab = -da_xb-da_sbsaxb-da_sbsasbsaxb
q_ab_ab_ababab = Symbol('q_ab_ab_ababab')
q_ab_ba_ababab = Symbol('q_ab_ba_ababab')
q_ab_aba_ababab = Symbol('q_ab_aba_ababab')
q_ab_bab_ababab= Symbol('q_ab_bab_ababab')
q_ab_abab_ababab = Symbol('q_ab_abab_ababab')
q_ab_baba_ababab = Symbol('q_ab_baba_ababab')
q_ab_ababa_ababab = 0
q_ab_babab_ababab = 0
q_ba_ba_ababab = Symbol('q_ba_ba_ababab')
q_ba_aba_ababab = Symbol('q_ba_aba_ababab')
q_ba_bab_ababab = Symbol('q_ba_bab_ababab')
q_ba_abab_ababab = Symbol('q_ba_abab_ababab')
q_ba_baba_ababab = Symbol('q_ba_baba_ababab')
q_ba_ababa_ababab = 0
q_ba_babab_ababab = 0
q_aba_aba_ababab = Symbol('q_aba_aba_ababab')
q_aba_bab_ababab = Symbol('q_aba_bab_ababab')
q_aba_abab_ababab = Symbol('q_aba_abab_ababab')
q_aba_baba_ababab = Symbol('q_aba_baba_ababab')
q_aba_ababa_ababab = 0
q_aba_babab_ababab = 0
q_bab_bab_ababab = Symbol('q_bab_bab_ababab')
q_bab_abab_ababab = Symbol('q_bab_abab_ababab')
q_bab_baba_ababab = Symbol('q_bab_baba_ababab')
```

```
q_bab_ababa_ababab = 0
q_bab_babab_ababab = 0
q_abab_abab_ababab = Symbol('q_abab_abab_ababab')
q_abab_baba_ababab = Symbol('q_abab_baba_ababab')
q_abab_ababa_ababab = 0
q_abab_babab_ababab = 0
q_baba_baba_ababab = Symbol('q_baba_baba_ababab')
q_baba_ababa_ababab = 0
q_baba_babab_ababab = 0
q_ababa_ababa_ababab = 0
q_ababa_babab_ababab = 0
q_babab_babab_ababab = 0

print("XaXa=", Symbol.simplify(q_a_a_ab*Xab+q_a_a_ba*Xba+q_a_a_aba*Xaba+q_a_a_bab*Xbab
              +q_a_a_abab*Xabab+q_a_a_baba*Xbaba+q_a_a_ababa*Xababa+q_a_a_babab*Xbabab+
              q_a_a_ababab*Xababab))
print("XaXb=", Symbol.simplify(q_a_b_ab*Xab+q_a_b_ba*Xba+q_a_b_aba*Xaba+q_a_b_bab*Xbab+
               q_a_b_abab*Xabab+q_a_b_baba*Xbaba+q_a_b_ababa*Xababa+q_a_b_babab*Xbabab+
               q_a_b_ababab*Xababab))
print("XaXab=", Symbol.simplify(q_a_ab_aba*Xaba+q_a_ab_bab*Xbab+q_a_ab_abab*Xabab+
                q_a_ab_baba*Xbaba+q_a_ab_ababa*Xababa+q_a_ab_babab*Xbabab+q_a_ab_ababab*Xababab))
print("XaXba=", Symbol.simplify(q_a_ba_aba*Xaba+q_a_ba_bab*Xbab+q_a_ba_abab*Xabab+q_a_ba_baba*Xbaba+
                q_a_ba_ababa*Xababa+q_a_ba_babab*Xbabab+q_a_ba_ababab*Xababab))
print("XaXaba=", Symbol.simplify(q_a_aba_abab*Xabab+q_a_aba_baba*Xbaba+q_a_aba_ababa*Xababa+
                 q_a_aba_babab*Xbabab+q_a_aba_ababab*Xababab))
print("XaXbab=", Symbol.simplify(q_a_bab_abab*Xabab+q_a_bab_baba*Xbaba+q_a_bab_ababa*Xababa+
                 q_a_bab_babab*Xbabab+q_a_bab_ababab*Xababab))
print("XaXabab=", Symbol.simplify(q_a_abab_ababa*Xababa+q_a_abab_babab*Xbabab+
                  q_a_abab_ababab*Xababab))
print("XaXbaba=", Symbol.simplify(q_a_baba_ababa*Xababa+q_a_baba_babab*Xbabab+
                  q_a_baba_ababab*Xababab))
print("XaXababa=", Symbol.simplify(q_a_ababa_ababab*Xababab))
print("XaXbabab=", Symbol.simplify(q_a_babab_ababab*Xababab))
print("XaXababab=", 0)
print("XbXb=", Symbol.simplify(q_b_b_ab*Xab+q_b_b_ba*Xba+q_b_b_aba*Xaba+q_b_b_bab*Xbab+
               q_b_b_abab*Xabab+q_b_b_baba*Xbaba+q_b_b_ababa*Xababa+q_b_b_babab*Xbabab+
               q_b_b_ababab*Xababab))
print("XbXab=", Symbol.simplify(q_b_ab_aba*Xaba+q_b_ab_bab*Xbab+q_b_ab_abab*Xabab+q_b_ab_baba*Xbaba+
                q_b_ab_ababa*Xababa+q_b_ab_babab*Xbabab+q_b_ab_ababab*Xababab))
print("XbXba=", Symbol.simplify(q_b_ba_aba*Xaba+q_b_ba_bab*Xbab+q_b_ba_abab*Xabab+q_b_ba_baba*Xbaba+
                q_b_ba_ababa*Xababa+q_b_ba_babab*Xbabab+q_b_ba_ababab*Xababab))
print("XbXaba=", Symbol.simplify(q_b_aba_abab*Xabab+q_b_aba_baba*Xbaba+q_b_aba_ababa*Xababa+
                q_b_aba_babab*Xbabab+q_b_aba_ababab*Xababab))
print("XbXbab=", Symbol.simplify(q_b_bab_abab*Xabab+q_b_bab_baba*Xbaba+q_b_bab_ababa*Xababa+
                q_b_bab_babab*Xbabab+q_b_bab_ababab*Xababab))
print("XbXabab=", Symbol.simplify(q_b_abab_ababa*Xababa+q_b_abab_babab*Xbabab+
                 q_b_abab_ababab*Xababab))
print("XbXbaba=", Symbol.simplify(q_b_baba_ababa*Xababa+q_b_baba_babab*Xbabab+
                  q_b_baba_ababab*Xababab))
print("XbXababa=", Symbol.simplify(q_b_ababa_ababab*Xababab))
print("XbXbabab=", Symbol.simplify(q_b_babab_ababab*Xababab))
print("XbXababab=", 0)
print("XabXab=", Symbol.simplify(q_ab_ab_aba*Xaba+q_ab_ab_bab*Xbab+q_ab_ab_abab*Xabab+
                 q_ab_ab_baba*Xbaba+q_ab_ab_ababa*Xababa+q_ab_ab_babab*Xbabab+q_ab_ab_ababab*Xababab))
print("XabXba=", Symbol.simplify(q_ab_ba_aba*Xaba+q_ab_ba_bab*Xbab+q_ab_ba_abab*Xabab+
                 q_ab_ba_baba*Xbaba+q_ab_ba_ababa*Xababa+q_ab_ba_babab*Xbabab+q_ab_ba_ababab*Xababab))
print("XabXaba=", Symbol.simplify(q_ab_aba_abab*Xabab+q_ab_aba_baba*Xbaba+q_ab_aba_ababa*Xababa+
                 q_ab_aba_babab*Xbabab+q_ab_aba_ababab*Xababab))
print("XabXbab=", Symbol.simplify(q_ab_bab_abab*Xabab+q_ab_bab_baba*Xbaba+q_ab_bab_ababa*Xababa+
                 q_ab_bab_babab*Xbabab+q_ab_bab_ababab*Xababab))
print("XabXabab=", Symbol.simplify(q_ab_abab_ababa*Xababa+q_ab_abab_babab*Xbabab+
                   q_ab_abab_ababab*Xababab))
```

```
print("XabXbaba=", Symbol.simplify(q_ab_baba_ababa*Xababa+q_ab_baba_babab*Xbabab+
                    q_ab_baba_ababab*Xababab))
print("XabXababa=", Symbol.simplify(q_ab_ababa_ababab*Xababab))
print("XabXbabab=", Symbol.simplify(q_ab_babab_ababab*Xababab))
print("XabXababab=", 0)
print("XbaXba=", Symbol.simplify(q_ba_ba_aba*Xaba+q_ba_ba_bab*Xbab+q_ba_ba_abab*Xabab+
                 q_ba_ba_baba*Xbaba+q_ba_ba_ababa*Xababa+q_ba_ba_babab*Xbabab+q_ba_ba_ababab*Xababab))
print("XbaXaba=", Symbol.simplify(q_ba_aba_abab*Xabab+q_ba_aba_baba*Xbaba+q_ba_aba_ababa*Xababa+
                 q_ba_aba_babab*Xbabab+q_ba_aba_ababab*Xababab))
print("XbaXbab=", Symbol.simplify(q_ba_bab_abab*Xabab+q_ba_bab_baba*Xbaba+q_ba_bab_ababa*Xababa+
                  q_ba_bab_babab*Xbabab+q_ba_bab_ababab*Xababab))
print("XbaXabab=", Symbol.simplify(q_ba_abab_ababa*Xababa+q_ba_abab_babab*Xbabab+
                   q_ba_abab_ababab*Xababab))
print("XbaXbaba=", Symbol.simplify(q_ba_baba_ababa*Xababa+q_ba_baba_babab*Xbabab+
                   q_ba_baba_ababab*Xababab))
print("XbaXababa=", Symbol.simplify(q_ba_ababa_ababab*Xababab))
print("XbaXbabab=", Symbol.simplify(q_ba_babab_ababab*Xababab))
print("XbaXababab=", 0)
print("XabaXaba=", Symbol.simplify(q_aba_aba_abab*Xabab+q_aba_aba_baba*Xbaba+q_aba_aba_ababa*Xababa+
                   q_aba_aba_babab*Xbabab+q_aba_aba_ababab*Xababab))
print("XabaXbab=", Symbol.simplify(q_aba_bab_abab*Xabab+q_aba_bab_baba*Xbaba+q_aba_bab_ababa*Xababa+
                  q_aba_bab_babab*Xbabab+q_aba_bab_ababab*Xababab))
print("XabaXabab=", Symbol.simplify(q_aba_abab_ababa*Xababa+q_aba_abab_babab*Xbabab+
                    q_aba_abab_ababab*Xababab))
print("XabaXbaba=", Symbol.simplify(q_aba_baba_ababa*Xababa+q_aba_baba_babab*Xbabab+
                    q_aba_baba_ababab*Xababab))
print("XabaXababa=", Symbol.simplify(q_aba_ababa_ababab*Xababab))
print("XabaXbabab=", Symbol.simplify(q_aba_babab_ababab*Xababab))
print("XabaXababab=", 0)
print("XbabXbab=", Symbol.simplify(q_bab_bab_abab*Xabab+q_bab_bab_baba*Xbaba+
                   q_bab_bab_ababa*Xababa+q_bab_bab_babab*Xbabab+q_bab_bab_ababab*Xababab))
print("XbabXabab=", Symbol.simplify(q_bab_abab_ababa*Xababa+q_bab_abab_babab*Xbabab+
                    q_bab_abab_ababab*Xababab))
print("XbabXbaba=", Symbol.simplify(q_bab_baba_ababa*Xababa+q_bab_baba_babab*Xbabab+
                    q_bab_baba_ababab*Xababab))
print("XbabXababa=", Symbol.simplify(q_bab_ababa_ababab*Xababab))
print("XbabXbabab=", Symbol.simplify(q_bab_babab_ababab*Xababab))
print("XbabXababab=", 0)
print("XababXabab=", Symbol.simplify(q_abab_abab_ababa*Xababa+q_abab_abab_babab*Xbabab+
                     q_abab_abab_ababab*Xababab))
print("XababXbaba=", Symbol.simplify(q_abab_baba_ababa*Xababa+q_abab_baba_babab*Xbabab+
                     q_abab_baba_ababab*Xababab))
print("XababXababa=", Symbol.simplify(q_abab_ababa_ababab*Xababab))
print("XababXbabab=", Symbol.simplify(q_abab_babab_ababab*Xababab))
print("XababXababab=", 0)
print("XbabaXbaba=", Symbol.simplify(q_baba_baba_ababa*Xababa+q_baba_baba_babab*Xbabab+
                     q_baba_baba_ababab*Xababab))
print("XbabaXababa=", Symbol.simplify(q_baba_ababa_ababab*Xababab))
print("XbabaXbabab=", Symbol.simplify(q_baba_babab_ababab*Xababab))
print("XbabaXababab=", 0)
print("XababaXababa=", Symbol.simplify(q_ababa_ababa_ababab*Xababab))
print("XababaXbabab=", 0)
print("XababaXababab=", 0)
print("XbababXbabab=", Symbol.simplify(q_babab_babab_ababab*Xababab))
print("XbababXababab=", 0)
print("XabababXababab=", 0)
```

Bibliography

[1] M. F. Atiyah and I. G. Macdonald. *Introduction to commutative algebra*. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1969.

[2] A. Björner and F. Brenti. *Combinatorics of Coxeter groups*, volume 231 of *Graduate Texts in Mathematics*. Springer, New York, 2005. doi:10.1007/3-540-27596-7.

[3] A. Borel. *Linear algebraic groups*. Notes taken by Hyman Bass. W. A. Benjamin, Inc., New York-Amsterdam, 1969. doi:10.1007/978-1-4612-0941-6.

[4] N. Bourbaki. *General topology. Chapters 1–4*. Elements of Mathematics (Berlin). Springer-Verlag, Berlin, 1989. Translated from the French, Reprint of the 1966 edition.

[5] N. Bourbaki. *General topology. Chapters 5–10*. Elements of Mathematics (Berlin). Springer-Verlag, Berlin, 1989. Translated from the French, Reprint of the 1966 edition.

[6] N. Bourbaki. *Lie groups and Lie algebras. Chapters 7–9*. Elements of Mathematics (Berlin). Springer-Verlag, Berlin, 2005. Translated from the 1975 and 1982 French originals by Andrew Pressley.

[7] B. Calmès, V. Petrov, and K. Zainoulline. Invariants, torsion indices and oriented cohomology of complete flags. *Ann. Sci. Éc. Norm. Supér. (4)*, 46(3):405–448 (2013), 2013. doi:10.24033/asens.2192. arXiv:0905.1341.

[8] B. Calmès, K. Zainoulline, and C. Zhong. A coproduct structure on the formal affine Demazure algebra. *Math. Z.*, 282(3-4):1191–1218, 2016. doi:10.1007/s00209-015-1585-6. arXiv:1209.1676.

[9] J. Cederstrand. StackOverflow Question 49670868. https://stackoverflow.com/questions/49670868/power-set-manipulation-in-python.

[10] M. Demazure. Invariants symétriques entiers des groupes de Weyl et torsion. *Invent. Math.*, 21:287–301, 1973. doi:10.1007/BF01418790.

[11] H. Duan. Schubert calculus and cohomology of Lie groups. Part II. Compact Lie groups. arXiv Preprint. arXiv:1502.00410.

[12] H. Duan and X. Zhao. Schubert calculus and cohomology of Lie groups. Part I. 1-connected lie groups. arXiv Preprint. arXiv:0711.2541.

[13] H. Duan and X. Zhao. Schubert presentation of the cohomology ring of flag manifolds G/T. *LMS J. Comput. Math.*, 18(1):489–506, 2015. doi:10.1112/S1461157015000133. arXiv:0801.2444.

[14] D. S. Dummit and R. M. Foote. *Abstract algebra*. Prentice Hall, Inc., Englewood Cliffs, NJ, 1991.

[15] W. Fulton. *Intersection theory*, volume 2 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge. A Series of Modern Surveys in Mathematics [Results in Mathematics and Related Areas. 3rd Series. A Series of Modern Surveys in Mathematics]*. Springer-Verlag, Berlin, second edition, 1998. doi:10.1007/978-1-4612-1700-8.

[16] R. Gandhi. Python scripts. URL: github.com/RajGandhi97/Oriented-cohomology-rings-of-the-split-semisimple-linear-algebraic-groups-of-rank-2.

[17] S. Gille and K. Zainoulline. Equivariant pretheories and invariants of torsors. *Transform. Groups*, 17(2):471–498, 2012. doi:10.1007/s00031-012-9178-5. arXiv:1007.3780.

[18] R. Hartshorne. *Algebraic geometry*. Springer-Verlag, New York-Heidelberg, 1977. Graduate Texts in Mathematics, No. 52.

[19] M. Hazewinkel. *Formal groups and applications*, volume 78 of *Pure and Applied Mathematics*. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York-London, 1978.

[20] A. Hoffnung, J. Malagón-López, A. Savage, and K. Zainoulline. Formal Hecke algebras and algebraic oriented cohomology theories. *Selecta Math. (N.S.)*, 20(4):1213–1245, 2014. doi:10.1007/s00029-013-0132-8. arXiv:1208.4114.

[21] J. E. Humphreys. *Introduction to Lie algebras and representation theory*. Springer-Verlag, New York-Berlin, 1972. Graduate Texts in Mathematics, Vol. 9.

[22] J. E. Humphreys. *Linear algebraic groups*. Springer-Verlag, New York-Heidelberg, 1975. Graduate Texts in Mathematics, No. 21.

[23] J. E. Humphreys. *Reflection groups and Coxeter groups*, volume 29 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1990. doi:10.1017/CBO9780511623646.

[24] T. Husain. *Topology and maps*, volume 5 of *Mathematical Concepts and Methods in Science and Engineering*. Plenum Press, New York-London, 1977. doi:10.1007/978-1-4615-8798-9.

[25] V. G. Kac. Torsion in cohomology of compact Lie groups and Chow rings of reductive algebraic groups. *Invent. Math.*, 80(1):69–79, 1985. doi:10.1007/BF01388548.

[26] C. Kassel. *Quantum groups*, volume 155 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995. doi:10.1007/978-1-4612-0783-2.

[27] M. A. Knus, A. Merkurjev, M. Rost, and J. P. Tignol. *The book of involutions*, volume 44 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Providence, RI, 1998. With a preface in French by J. Tits.

[28] M. Lanini and K. Zainoulline. Twisted quadratic foldings of root systems. *Algebra i Analiz*, 33(1):93–118, 2021. arXiv:1806.08962.

[29] T. Leinster. *Basic category theory*, volume 143 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2014. doi:10.1017/CBO9781107360068.

[30] M. Levine. The algebraic K-theory of the classical groups and some twisted forms. *Duke Math. J.*, 70(2):405–443, 1993. doi:10.1215/S0012-7094-93-07008-1.

[31] M. Levine and F. Morel. *Algebraic cobordism*. Springer Monographs in Mathematics. Springer, Berlin, 2007. doi:10.1007/3-540-36824-8.

[32] J. R. Munkres. *Topology: a first course*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1975.

[33] J. Nagata. *Modern general topology*, volume 33 of *North-Holland Mathematical Library*. North-Holland Publishing Co., Amsterdam, second edition, 1985.

[34] H. V. Pittie. The integral homology and cohomology rings of $\mathrm{SO}(n)$ and $\mathrm{Spin}(n)$. *J. Pure Appl. Algebra*, 73(2):105–153, 1991. doi:10.1016/0022-4049(91)90108-E.

[35] I. R. Shafarevich. *Basic algebraic geometry. 1.* Springer-Verlag, Berlin, second edition, 1994. Varieties in projective space, Translated from the 1988 Russian edition and with notes by Miles Reid.

[36] I. R. Shafarevich. *Basic algebraic geometry. 2.* Springer-Verlag, Berlin, second edition, 1994. Schemes and complex manifolds, Translated from the 1988 Russian edition by Miles Reid.

[37] K. Smith. Algebraic Geometry. Online Lecture Notes. 2012. http://www.math.lsa.umich.edu/~kesmith/631Lectures.pdf.

[38] K. E. Smith, L. Kahanpää, P. Kekäläinen, and W. Traves. *An invitation to algebraic geometry.* Universitext. Springer-Verlag, New York, 2000. doi:10.1007/978-1-4757-4497-2.

[39] T. A. Springer. *Linear algebraic groups*, volume 9 of *Progress in Mathematics.* Birkhäuser Boston, Inc., Boston, MA, second edition, 1998. doi:10.1007/978-0-8176-4840-4.

[40] B. Totaro. The torsion index of E_8 and other groups. *Duke Math. J.*, 129(2):219–248, 2005. doi:10.1215/S0012-7094-05-12922-2.

[41] B. Totaro. The torsion index of the spin groups. *Duke Math. J.*, 129(2):249–290, 2005. doi:10.1215/S0012-7094-05-12923-4.

[42] R. G. Underwood. *An introduction to Hopf algebras.* Springer, New York, 2011. doi:10.1007/978-0-387-72766-0.

[43] C. A. Weibel. *The K-book*, volume 145 of *Graduate Studies in Mathematics.* American Mathematical Society, Providence, RI, 2013. An introduction to algebraic K-theory.

[44] N. Yagita. Algebraic cobordism of simply connected Lie groups. *Math. Proc. Cambridge Philos. Soc.*, 139(2):243–260, 2005. doi:10.1017/S0305004105008510.

Index