# PARTITION IDENTITIES ARISING FROM THE STANDARD $A_2^{(2)}$-MODULES OF LEVEL 4

**BY DEBAJYOTI NANDI**

**A dissertation submitted to the**

**Graduate School—New Brunswick**

**Rutgers, The State University of New Jersey**

**in partial fulfillment of the requirements**

**for the degree of**

**Doctor of Philosophy**

**Graduate Program in Mathematics**

**Written under the direction of**

**Robert L. Wilson**

**and approved by**

**New Brunswick, New Jersey**

**October, 2014**

## ABSTRACT OF THE DISSERTATION

# Partition Identities Arising from the Standard $A_2^{(2)}$-Modules of Level 4

**by Debajyoti Nandi**

**Dissertation Director: Robert L. Wilson**

In this dissertation, we propose a set of new partition identities, arising from a twisted vertex operator construction of the level 4 standard modules for the affine Kac-Moody algebra of type $A_2^{(2)}$. These identities have an interesting new feature, absent from previously known examples of this type.

This work is a continuation of a long line of research of constructing standard modules for affine Kac-Moody algebras via vertex operators, and the associated combinatorial identities. The interplay between representation theory and combinatorial identities was exemplified by the vertex-algebraic proof of the famous Rogers-Ramanujan-type identities using standard $A_1^{(1)}$-modules by J. Lepowsky and R. Wilson. In his Ph.D. thesis, S. Capparelli proposed new combinatorial identities using a twisted vertex operator construction of the standard $A_2^{(2)}$-modules of level 3, which were later proved independently by G. Andrews, S. Capparelli, and M. Tamba-C. Xie.

We begin with an obvious spanning set for each of the level 4 standard modules for $A_2^{(2)}$, and reduce this spanning set using various relations. Most of these relations come from certain product generating function identities which are valid for all the level 4 modules. There are also other ad-hoc relations specific to a particular module of level 4. In this way, we reduce our spanning sets to match with the graded dimensions of the

said modules as closely as possible. We conjecture and present strong evidence for three partition identities based on the spanning sets for the three standard $A_2^{(2)}$-modules of level 4.

One surprising result of our work is the discovery of relations of arbitrary length. Consequently, the partitions corresponding to these spanning sets cannot be described by difference conditions of finite length.

The spanning set result proves one inequality of the proposed identities. There is strong evidence for the validity of the conjecture (i.e., the opposite inequality), since it has been verified to hold for partitions of $n \leq 170$, and $n = 180, 190$ and 200.

# Acknowledgements

This work would not have been possible without the support and encouragement of many people. But, foremost, I wish to express my sincerest gratitude to my adviser, Prof. Robert Wilson, for his continued support, patience and valuable guidance. I would also like to thank Prof. James Lepowsky his guidance and insightful comments during the review of this thesis. I would like to thank my colleague and friend Shashank Kanade for reading this thesis and suggesting improvements, and Matthew Russell for his help with the Maple codes for computations in noncommutative algebras.

I am also grateful to my friends Amit, Aravind, Bibhash, Chitra, Moulik, Samreen, Supriya, Swathi, Yevgen, and many others for all their support, patience and the fun times together.

I am specially indebted to my family—my mother Sulata Nandi, my sister, Soumi Nandi, and my brother-in-law Anirban Biswas—for their unwavering love, support, understanding and encouragement.

# Dedication

To my loving mother

**Sulata Nandi**,

my lovely sister

**Soumi Nandi**,

and in memory of my father

**Late Joydeb Nandi**.

# Table of Contents

# Chapter 1

# Introduction

In this work, we conjecture and present strong evidence for possible partition identities arising from the standard modules of level 4 for the affine Lie algebra $\mathrm{A}_2^{(2)}$ using a twisted vertex operator construction. Historically, the discovery of vertex operator constructions of representations of affine Lie algebras was motivated by a conjectured interplay between classical partition identities and standard modules for affine Kac-Moody Lie algebras.

The first famous example of such interplay arises from the Rogers-Ramanujan identities, which may be stated as follows:

(i) *The number of partitions of a nonnegative integer $n$ in which the difference between any two successive parts is at least 2 is the same as the number of partitions of $n$ into parts congruent to 1 or 4 modulo 5.*

(ii) *The number of partitions of a nonnegative integer $n$ in which the difference between any two successive parts is at least 2 and such that the smallest part is at least 2 is the same as the number of partitions of $n$ into parts congruent to 2 or 3 modulo 5.*

A connection between the congruence conditions and standard modules for $\mathrm{A}_1^{(1)}$ was discovered by J. Lepowsky and S. Milne [LM78]. A vertex operator theoretic interpretation and proof of the Rogers-Ramanujan identities, "explaining" the difference conditions, was given by Lepowsky and R. Wilson in [LW82, LW84]. They used monomials, acting on a highest weight vector, in certain new operators whose indices reflected the difference conditions. They extended their work to all the standard $\mathrm{A}_1^{(1)}$-modules in [LW82, LW84, LW85], giving a vertex-algebraic interpretation of a family of Rogers-Ramanujan-type identities, discovered by B. Gordon, G. Andrews and D. Bressoud. The case for the level 2 standard $A_1^{(1)}$-modules was described by certain "difference-one" conditions (where adjacent parts have difference at least one). For level 3 it was described by "difference-two" conditions (where adjacent parts have difference at least two). For levels greater than 3, the description changed into "difference-two-at-a-distance" and parity conditions, reflecting the sum sides of the Gordon-Andrews-Bressoud identities.

The linear independence of the relevant monomials (applied to a highest weight vector) for standard $A_1^{(1)}$-modules of level greater than 3 was not proved in the sequel [LW82, LW84, LW85]. This problem was solved by A. Meurman and M. Primc [MP87], providing a vertex-algebraic proof of the Gordon-Andrews-Bressoud identities beyond the case of Rogers-Ramanujan identities.

In his Ph.D. thesis [Cap88], S. Capparelli proposed a pair of combinatorial identities based on the standard $A_2^{(2)}$-modules of level 3. He also demonstrated that the construction of the level 2 standard modules for $A_2^{(2)}$ in this way gives rise to another vertex operator theoretic interpretation of the classical Rogers-Ramanujan identities (see also [Cap92, Cap93]). It was believed that once a few low level cases for standard $A_2^{(2)}$-modules had been successfully analyzed in this way, a general construction for all levels would emerge. However, the cases for $A_2^{(2)}$ turned out to be much harder and subtler than those for $A_1^{(1)}$ which had been extensively studied. One of Capparelli's identities, arising from the level 3 standard $A_2^{(2)}$-modules, may be stated as follows:

> *The number of partitions of a nonnegative integer $n$ into parts different from $1$ and such that the difference of two successive parts is at least $2$, and is exactly $2$ or $3$ only if their sum is a multiple of $3$, is the same as the number of partitions of $n$ into parts congruent to $\pm 2, \pm 3$ modulo $12$.*

A $q$-series proof of this identity was given by G. Andrews [And94], proving Capparelli's conjecture. Capparelli also provided a direct vertex operator theoretic proof of his identities by proving the linear independence of his spanning sets in [Cap96]. M. Tamba and C. Xie [TX95] independently gave another vertex operator theoretic proof of Capparelli's identities. See [Lep07] for more details.

In this work, we give combinatorial interpretations of the graded dimensions of the three inequivalent standard $A_2^{(2)}$-modules of level 4. The level 4 case turns out to be much more difficult and subtle even compared to the level 3 case, showing even more surprising results.

A partition can be thought of as a non-increasing sequence of positive integers. A partition $(m_1, \ldots, m_s)$ is said to satisfy a difference condition $[d_1, \ldots, d_{s-1}]$ if $m_i - m_{i+1} = d_i$ for all $1 \leq i \leq s-1$. The partition identities we propose, based on the three inequivalent standard $A_2^{(2)}$-modules of level 4, may be stated as follows:

(i) *The number of partitions of a nonnegative integer $n$ into parts different from 1 and such that there is no sub-partition satisfying the difference conditions $[1]$, $[0,0]$, $[0,2]$, $[2,0]$ or $[0,3]$, and such that there is no sub-partition with an odd sum of parts satisfying the difference conditions $[3,0]$, $[0,4]$, $[4,0]$ or $[3,2^*,3,0]$ (where $2^*$ indicates zero or more occurrence of 2), is the same as the number of partitions of $n$ into parts congruent to $\pm 2, \pm 3$ or $\pm 4$ modulo 14.*

(ii) *The number of partitions of a nonnegative integer $n$ such that 1, 2 and 3 may occur at most once as a part, and such that there is no sub-partition satisfying the difference conditions $[1]$, $[0,0]$, $[0,2]$, $[2,0]$ or $[0,3]$, and such that there is no sub-partition with an odd sum of parts satisfying the difference conditions $[3,0]$, $[0,4]$, $[4,0]$ or $[3,2^*,3,0]$ (where $2^*$ indicates zero or more occurrence of 2), is the same as the number of partitions of $n$ into parts congruent to $\pm 1, \pm 4$ or $\pm 6$ modulo 14.*

(iii) *The number of partitions of a nonnegative integer $n$ into parts different from 1 and 3, such that 2 may occur at most once as a part, and such that there is no sub-partition satisfying the difference condition $[3,2^*]$ (where $2^*$ denotes zero or more occurrence of 2) ending with a 2, and such that there is no sub-partition satisfying the difference conditions $[1]$, $[0,0]$, $[0,2]$, $[2,0]$ or $[0,3]$, and such that there is no sub-partition with an odd sum of parts satisfying the difference conditions $[3,0]$, $[0,4]$, $[4,0]$ or $[3,2^*,3,0]$, is the same as the number of partitions of $n$ into parts congruent to $\pm 2, \pm 5$ or $\pm 6$*

*modulo* 14.

Each of the above statements corresponds to computing the graded dimension of a level 4 standard $A_2^{(2)}$-module in two ways—from the principal specialization of the Weyl-Kac character formula given by the numerator formula (see [LM78, Lep78]) (describing the "congruence conditions") and an explicit construction of a graded basis for the module (describing the "difference conditions" and "initial conditions").

The graded dimension, given by the principal specialization of the Weyl-Kac character formula and the numerator formula of [LM78, Lep78], can be factored as $\chi(q) = H(q)F(q)$, as a formal power series in $q$, where $F(q)$ is the series that counts the partitions with the "congruence conditions." The extra factor $H(q)$ is similar to the "fudge factor" in [LM78, LW82, LW84, LW85]. In their works, Lepowsky and Wilson used a certain "vacuum space" and certain "$Z$-operators" to cancel out the "fudge factor." We show an equivalent cancellation without using such a "vacuum space."

In this dissertation, we prove the appropriate "spanning set" result. The starting point is a certain obvious spanning set, parametrized by two sets of partitions. The elements of this spanning set can be described as products of two types of operators—the "negative Heisenberg operators" and the "$X(\bullet)$ operators"—acting on a highest weight vector $v_0$. The partitions describe the degrees and the order of these operators applied to $v_0$. We show that no relations among the "negative Heisenberg operators" exist and that these operators are accounted for by the "fudge factor" $H(q)$. The only relations, therefore, come from the relations among the $X(\bullet)$ operators acting on $v_0$.

We eliminate extraneous elements from this spanning set based on these relations. The resulting pruned spanning set can be described as parametrized by the set of all partitions which do not contain certain "forbidden" sub-partitions. The most surprising result in our work was the discovery of forbidden sub-partitions of arbitrary lengths. These forbidden partitions can be described by the "difference conditions" mentioned above. In all previously known analogous situations arising from representations of affine Kac-Moody algebras, the forbidden partitions could be described by difference conditions of bounded length. For example, in the Rogers-Ramanujan identities and Capparelli's identities, the difference conditions are of length one (reflecting the difference between adjacent parts). In our case, there are forbidden partitions satisfying arbitrarily long difference conditions. These difference conditions are the same for the all standard $A_2^{(2)}$-modules of level 4. The differentiating factors are then the "initial conditions" associated with the three inequivalent level 4 standard modules.

If the resulting spanning set is linearly independent, then the product side given by $F(q)$ may be expressed as $\sum_{n\geq 0} A_n q^n$, where $A_n$ is the number of partitions of $n$ not containing any forbidden sub-partitions. We call these partitions "allowed" partitions.

Our spanning set result states that in each of the above cases, the number of partitions of $n$ described by various initial and difference conditions is greater than or equal to the number of partitions of $n$ into parts satisfying the corresponding modulo 14 conditions. Experimental evidence shows that the equality holds for $n \leq 170$, as well as for $n = 180, 190$ and 200.

It is interesting to note how we discovered the family of "exceptional" forbidden partitions of arbitrary lengths (i.e., partitions of an odd number satisfying the difference conditions $[4, 0]$, $[3, 2^*, 3, 0]$). We set out to compare the graded dimension of the $(4, 0)$-module (one of the level 4 standard modules), with the spanning set we got after eliminating partitions into parts different from 1 (the initial condition for this module), and the other partitions containing forbidden sub-partitions, using relations similar to what Capparelli used in [Cap88, Cap93]. We found the first discrepancy at $n = 13$, and the next one at $n = 19$. In each case, there was an extra partition in our pruned spanning set compared to what the corresponding graded dimension would suggest. From certain "periodicity properties" of our relations, we could infer that we must have missed a forbidden triplet (partition into 3 parts). The smallest such triplet surviving in our spanning set (for $n = 13$) was $(7, 3, 3)$. We then eliminated $(7, 3, 3)$ and all its 2-translates (i.e., partitions of the form $(7 + 2k, 3 + 2k, 3 + 2k)$, $k \geq 0$). We compared our resulting spanning set with the graded dimension again, and noticed that the next two discrepancies were at $n = 21$ and $n = 29$, and in each case there was one extra partition in our spanning set. Once again, the "periodicity properties" suggested that we must have missed a forbidden quadruplet (i.e., a partition into 4 parts). Eliminating the smallest surviving quadruplet and its 2-translates gave us a contradiction, i.e., we got a smaller number of partitions in the spanning set than required by the graded dimension, for $n$ sufficiently large. Therefore, we proceeded to eliminate the second quadruplet, which was $(9, 6, 3, 3)$, and its 2-translates. Proceeding in similar fashion, a clear pattern emerged for the family of forbidden partitions of arbitrary lengths.

The task of proving that these partitions are indeed forbidden turned out to be very subtle. Unlike in [Cap88, Cap93], we needed to keep track of terms containing "positive Heisenberg elements" in the relations that we used. In the case of the level 3 standard modules, the forbidden partitions arose directly from certain generating function identities. In our case, we obtain "longer" relations by multiplying similar generating function identities by suitable operators. The "exceptional" forbidden partitions of arbitrary length arise from these relations. Also, the initial conditions are significantly more difficult for level 4 than for level 3.

As illustrated by all of these phenomena, the level 4 theory for $\mathrm{A}_2^{(2)}$ is much more complex than the level 3 theory.

Now we give a brief overview of this dissertation.

In Chapter 2, we recall the basic definitions and results to describe the twisted vertex operator construction of the principally graded realization of the algebra $\mathrm{A}_2^{(2)}$. This is a simplification of the general case, based on vertex operator calculus, described in [Lep85, Fig87, Cap92, Cap93, FLM87, FLM88, DL96], specialized to our specific case of $\mathrm{A}_2^{(2)}$.

In Chapter 3, we recall the basic notions about standard modules for an affine Lie algebra and show that any level 4 standard module can be thought of as embedded in the tensor product of 4 copies of the basic module. We also recall the graded dimensions of these modules given by the principal specialization of the Weyl-Kac character formula and the numerator formula (see [Lep78, LM78] for more details).

In Chapter 4, we present the framework—some notations, definitions and results—on which the rest of the dissertation depends. First, we present a few definitions, notations and results related to partitions and generalized partitions (i.e., any sequence of integers, not necessarily positive, in non-increasing order). Then we describe certain standard monomials—parametrized by these partitions and generalized partitions—in certain operators and the structure of the standard modules in terms of the action of these monomials on a highest weight vector. We also present a number of substantial tools and techniques that we use repeatedly in the later chapters.

In Chapter 5, we present the "product generating function" identities that hold in any level 4 standard module (more generally, on the tensor product of 4 copies of the basic module). These identities are analogous to those used in [Cap88, Cap92, Cap93] for the standard $A_2^{(2)}$-modules of level 3. We also present the coefficients of the standard monomials that appear in these "product generating function" identities.

Chapter 6 is devoted to finding forbidden partitions using the product generating function identities mentioned above. There are two types of forbidden partitions. Those that follow directly from the product generating function identities, similar to those in the level 3 case in [Cap88, Cap92, Cap93], are called "regular" forbidden partitions. Interestingly, there are other forbidden partitions of arbitrary length (starting from length 3) satisfying a simple pattern of difference conditions. There are no analogues of this type of forbidden partitions in any of the previous cases. We call them "exceptional" forbidden partitions. These exceptional forbidden partitions follow from new relations obtained by multiplying the product generating function identities by suitable operators.

In Chapter 7, we describe the "initial conditions" for each of the three inequivalent level 4 standard modules for $A_2^{(2)}$. These come from certain ad-hoc relations specific to each of the particular standard $A_2^{(2)}$-modules of level 4, needed to match the graded dimensions of "low degrees."

Finally, in Chapter 8, we summarize our main results and our three (conjectured) partition identities arising from the three level 4 standard $A_2^{(2)}$-modules.

Some of the computations used in the proofs were performed using computer programs in Maple. We also wrote a C (standard C99) program to verify the validity of our partition identities. We have collected all the programs that we used in the appendices.

In Appendix A, we present the Maple worksheet and the Maple source files that we used (mainly in Chapter 6 and Chapter 7) for the computations of the relations.

In Appendix B, we present our Maple source files for computations in noncommutative algebras. We also present two Maple worksheets showing the computations used in some of the proofs (notably, in Chapter 7). The Maple programs implementing the operations (addition, multiplication, etc..) in noncommutative algebras were based on the NCFPS (noncommutative formal power series) package of D. Zeilberger (see [Zei12, BRRZ12]). The algorithm to apply substitution rules to straighten out an out-of-order monomial is based on the Maple codes of M. Russell (see [Rus13]). His program was for a finite number of substitution rules over a finite alphabet. We modified his program to work with an infinite number of substitution rules (based on finitely many patterns) over an infinite indexed alphabet.

In Appendix C, we present our C program (written in C99 standard) to verify our partition identities up to $n \leq 200$. (Note that we have done the verification only for $n \leq 170$ and for $n = 180, 190$ and 200. It may take more than 24 hours to complete the computation for $n = 200$.) We used the “accelerated ascending rule” algorithm of J. Kelleher (see [Kel06]) to generate all partitions of a nonnegative integer $n$.

# Chapter 2

# Preliminaries

In this chapter, we will discuss briefly the principally graded realization of the affine Lie algebra $\mathrm{A}_2^{(2)}$ using twisted vertex operators. The general set-up has been described in [FLM88, FLM87] in a more general setting. We will also follow closely the notations used in [Cap88, Cap92, Cap93]. Here we present the case of $\mathrm{A}_2^{(2)}$, which is much simpler than the general case.

We start with the root lattice $\mathrm{A}_2$, and construct a central extension of this root lattice to describe $\mathrm{A}_2^{(2)}$. Since this extension splits, much of the complication that arises in the general construction can be simplified. For detailed description of the general construction see [FLM88, FLM87, Cap88, Cap92, Cap93, LW84, Lep78, Lep85, Fig87, DL96].

## 2.1 Formal Calculus

In this section, we will give a brief overview of the formal calculus used in this dissertation. We only quote a few results. For details and the proofs see [FLM88, Cap92, LL04].

Let $V$ be any vector space over $\mathbb{C}$. Denote by $(\operatorname{End} V)[[z, z^{-1}]]$ the space of formal Laurent series in $z$ with coefficients in $\operatorname{End} V$. The elements of $(\operatorname{End} V)[[z, z^{-1}]]$ are denoted using "function" notation:

$$f(z) = \sum_{n \in \mathbb{Z}} f_n z^{-n}.$$

**Definition 2.1.1.** A (possibly infinite) subset $S \subset \operatorname{End} V$ is called *summable* if the set $\{\, f \in S \mid \ fv \neq 0 \,\}$ is finite for all $v \in V$.

*Remark.* If $S \subset \operatorname{End} V$ is summable then $\sum_{f \in S} f$ is a well-defined operator on $V$.

Henceforth we assume that $V$ is graded, i.e., $V = \coprod_{n\in\mathbb{Z}} V_n$, where $V_n$ denotes the set of all homogeneous elements in $V$ of degree $n$.

**Definition 2.1.2.** An endomorphism $f \in \operatorname{End} V$ is called *homogeneous of degree $d$* if it maps elements of $V_n$ into $V_{n+d}$ for all $n \in \mathbb{Z}$.

**Proposition 2.1.3.** *Let $f(z), g(z) \in (\operatorname{End} V)[[z, z^{-1}]]$. Assume that*

(i) *$f_n$, $g_n$ are homogeneous operators of degree $n$ on $V$ for all $n \in \mathbb{Z}$,*

(ii) *$[f_n, g_m] = 0$ for all $n, m \in \mathbb{Z}$,*

(iii) *$V$ is bounded above (or, below), i.e., there exists $N \in \mathbb{Z}$ such that $V_n = 0$ for all $n > N$ (or, $n < N$).*

*Then $f(z)g(z)$ is a well-defined element of* $(\operatorname{End} V)[[z, z^{-1}]]$.

*Proof.* Assume that $V$ is bounded above with highest degree $N$. For each $k \in \mathbb{Z}$, the coefficient of $z^{-k}$ in $f(z)g(z)$ is $\sum_{n+m=k} f_n g_m$. For a homogeneous vector $v \in V$ of degree $d$, the sum

$$\sum_{n+m=k} f_n g_m v = \sum_{\substack{n+m=k \\ m\leq N-d}} f_n g_m v = \sum_{\substack{n+m=k \\ m,n\leq N-d}} g_m f_n v$$

has only finitely many nonzero terms.

The proof for the case where $V$ is bounded below is similar. □

We now recall the limit notation. Let $(\operatorname{End} V)[[z_1^{\pm 1}, \ldots, z_n^{\pm 1}]]$ denote the space of all formal Laurent series in commuting indeterminates $z_1, \ldots, z_n$ with coefficients in $\operatorname{End} V$. Write

$$f(z_1, \ldots, z_n) = \sum_{i_1,\ldots,i_n\in\mathbb{Z}} f(i_1, \ldots, i_n)\, z_1^{-i_1} \cdots z_n^{-i_n}.$$

**Definition 2.1.4.** Define the *limit as all the indeterminates are set to $z$* by

$$\lim_{z_i\to z} f(z_1, \ldots, z_n) = \sum_{k\in\mathbb{Z}} \left( \sum_{\substack{i_1,\ldots,i_n\in\mathbb{Z} \\ i_1+\cdots+i_n=k}} f(i_1, \ldots, i_n) \right) z^{-k},$$

provided the the family

$$\left\{ f(i_1, \ldots, i_n) \,\middle|\, i_1 + \cdots + i_n = k \right\}$$

is summable for all $k \in \mathbb{Z}$.

Following [FLM88], we observe the following convention:

$$(z_1 + z_2)^n = z_1^n \left(1 + \frac{z_2}{z_1}\right)^n,$$

where the latter factor is to be expanded as a binomial series. Of course, this matters only when $n$ is not a positive integer. For example,

$$\frac{1}{z_1 + z_2} = z_1^{-1} \left(1 - \frac{z_2}{z_1} + \left(\frac{z_2}{z_1}\right)^2 - \cdots\right). \tag{2.1.1}$$

We quote a few useful properties of the limit below. For proofs see [FLM88, Cap92, LL04].

**Proposition 2.1.5.**

(1) *Let $f = f(z_1, \ldots, z_m)$ be a formal Laurent series such that $\lim_{z_i \to z} f$ exists. If $P = P(z_1, \ldots, z_m)$ is a Laurent polynomial, then $\lim_{z_i \to z} Pf$ exists and*

$$\lim_{z_i \to z} Pf = \left(\lim_{z_i \to z} P\right) \left(\lim_{z_i \to z} f\right).$$

(2) *Let $f = f(z_1, \ldots, z_m, w_1, \ldots, w_n)$ be a formal Laurent series such that $\lim_{z_i, w_j \to z} f$ exists. Then*

$$\lim_{z_i, w_j \to z} f = \lim_{z_i \to z} \left(\lim_{w_j \to z} f\right).$$

**Proposition 2.1.6.** *Assume that $V$ is bounded above (or, below). Let $f(z_1)$, $g(z_2)$ be formal Laurent series in the two commuting indeterminates $z_1$ and $z_2$, such that the coefficients $f_n, g_n$ are homogeneous operators of degree $n$ on $V$. Let $p = p(z_1, z_2)$ be a Laurent polynomial with constant coefficients such that*

$$p(z_1, z_2)[f(z_1), g(z_2)] = 0.$$

*Then the limit*

$$\lim_{z_1, z_2 \to z} p(z_1, z_2) f(z_1) g(z_2)$$

*exists.*

We recall two very useful Laurent series with constant coefficients to be used later in our exposition. The first is the “delta function,”

$$\delta(z) = \sum_{n \in \mathbb{Z}} z^n. \tag{2.1.2}$$

Let D denote the differential operator $\mathrm{D} = z\frac{\mathrm{d}}{\mathrm{d}z}$. Then,

$$\mathrm{D}\,\delta(z) = \sum_{n\in\mathbb{Z}} nz^n. \tag{2.1.3}$$

We quote the following well known properties of $\delta(z)$ and $\mathrm{D}\,\delta(z)$.

**Proposition 2.1.7.** *Let $f(z)$ be any Laurent polynomial over any algebra over $\mathbb{C}$, and $a \in \mathbb{C}^\times$ be a nonzero constant. Then we have*

(i) *$f(z)\delta(z) = f(1)\delta(z)$, or more generally, $f(z)\delta(a^{-1}z) = f(a)\delta(a^{-1}z)$, and in particular, $(1-a^{-1}z)\delta(a^{-1}z) = 0$;*

(ii) *$f(z)\,\mathrm{D}\,\delta(z) = f(1)\,\mathrm{D}\,\delta(z) - (\mathrm{D}\,f)(1)\delta(z)$, or more generally,*

$$f(z)\,\mathrm{D}\,\delta(a^{-1}z) = f(a)\,\mathrm{D}\,\delta(a^{-1}z) - (\mathrm{D}\,f)(a)\delta(a^{-1}z),$$

*and in particular, $(1-a^{-1}z)^2\,\mathrm{D}\,\delta(a^{-1}z) = 0$.*

*Proof.* It can be easily proved on each monomial $z^k$ of $f(z)$. For details see [FLM88]. $\square$

## 2.2 Vertex Operators

In this section, we describe the vertex operators used in the construction of $\mathrm{A}_2^{(2)}$. We follow the general method as described in [FLM88, FLM87, Lep85, LW84, LW85, Cap88, Cap92, Cap93, Fig87, DL96], simplifying the process as applicable to the case of $\mathrm{A}_2^{(2)}$.

Let $\Phi$ be the $\mathrm{A}_2$ root system with basis $\Delta = \{\alpha_1, \alpha_2\}$. Let $L = \mathbb{Z}\alpha_1 + \mathbb{Z}\alpha_2$ be the root lattice of $\mathrm{A}_2$, equipped with a symmetric $\mathbb{Z}$-bilinear form $\langle\,\cdot\,,\,\cdot\,\rangle$, where $\langle\alpha_i, \alpha_i\rangle = 2$ for $i = 1, 2$, and $\langle\alpha_1, \alpha_2\rangle = -1$. Note that the angle between $\alpha_1$ and $\alpha_2$ is $\frac{2\pi}{3}$.

Let $\nu$ be the automorphism of $L$ of order 6 acting as a rotation on the root system by $\frac{\pi}{3}$. On the basis elements of $\Delta$, $\nu(\alpha_1) = \alpha_1 + \alpha_2$, $\nu(\alpha_2) = -\alpha_1$. Clearly, $\nu^6 = 1$, and $\langle\nu^3\alpha, \alpha\rangle = -\langle\alpha, \alpha\rangle \in 2\mathbb{Z}$ for all $\alpha \in L$. Note that

$$\sum_{p\in\mathbb{Z}_6} \nu^p\alpha = 0, \quad \text{for all } \alpha \in L. \tag{2.2.1}$$

Let $\mathfrak{h} = \mathbb{C} \bigotimes_{\mathbb{Z}} L$. The form $\langle\,\cdot\,,\,\cdot\,\rangle$ can be linearly extended to $\mathfrak{h}$. Let $\omega$ be a primitive sixth root of unity. For concreteness, we may choose

$$\omega = e^{i\pi/3} = \frac{1}{2} + \frac{\sqrt{3}}{2}i.$$

For $n \in \mathbb{Z}$, set

$$\mathfrak{h}_{(n)} = \left\{ x \in \mathfrak{h} \;\middle|\; \nu x = \omega^n x \right\}. \tag{2.2.2}$$

For $n \equiv m \pmod 6$, $\omega^n = \omega^m$ and $\mathfrak{h}_{(n)} = \mathfrak{h}_{(m)}$. Thus the expressions $\omega^p$ and $\mathfrak{h}_{(p)}$ have obvious well-defined meaning for $p \in \mathbb{Z}_6$. We have

$$\mathfrak{h} = \coprod_{p \in \mathbb{Z}_6} \mathfrak{h}_{(p)}. \tag{2.2.3}$$

Note that $\mathfrak{h}_{(n)} = 0$ unless $n \equiv \pm 1 \pmod 6$. Therefore,

$$\mathfrak{h} = \mathfrak{h}_{(1)} \bigoplus \mathfrak{h}_{(-1)} \tag{2.2.4}$$

is the eigenspace decomposition for the action of $\nu$ on $\mathfrak{h}$.

Viewing $\mathfrak{h}$ as an abelian Lie algebra, construct the $\nu$-twisted affine Lie algebra

$$\begin{aligned} \widetilde{\mathfrak{h}} = \widetilde{\mathfrak{h}}[\nu] &= \coprod_{n \in \mathbb{Z}} \left( \mathfrak{h}_{(n)} \otimes t^{n/6} \right) \oplus \mathbb{C}c \oplus \mathbb{C}d \\ &= \coprod_{n \in \frac{1}{6}\mathbb{Z}} \left( \mathfrak{h}_{(6n)} \otimes t^{n} \right) \oplus \mathbb{C}c \oplus \mathbb{C}d, \end{aligned} \tag{2.2.5}$$

with the following brackets:

$$\begin{gathered} [x \otimes t^{i/6},\, y \otimes t^{j/6}] = \tfrac{i}{6} \langle x, y \rangle \delta_{i+j,0} c, \\ [d,\, x \otimes t^{i/6}] = \tfrac{i}{6} x \otimes t^{i/6}, \\ [c,\, x \otimes t^{i/6}] = [c,\, d] = 0, \end{gathered} \tag{2.2.6}$$

for all $i, j \in \mathbb{Z}$, $x \in \mathfrak{h}_{(i)}$, $y \in \mathfrak{h}_{(j)}$.

Consider the commutator subalgebra $\mathfrak{s}$ of $\widetilde{\mathfrak{h}}[\nu]$

$$\mathfrak{s} = \coprod_{\substack{n \in \mathbb{Z} \\ n \neq 0}} \left( \mathfrak{h}_{(n)} \otimes t^{n/6} \right) \oplus \mathbb{C}c = \coprod_{\substack{n \in \mathbb{Z} \\ n \equiv \pm 1 \pmod 6}} \left( \mathfrak{h}_{(n)} \otimes t^{n/6} \right) \oplus \mathbb{C}c, \tag{2.2.7a}$$

and the subalgebras

$$\mathfrak{s}_{\pm} = \coprod_{\substack{n \in \mathbb{Z} \\ \pm n > 0}} \left( \mathfrak{h}_{(n)} \otimes t^{n/6} \right) \oplus \mathbb{C}c, \tag{2.2.7b}$$

$$\begin{aligned} \mathfrak{b} = \mathfrak{b}[\nu] &= \coprod_{\substack{n \in \mathbb{Z} \\ n \geq 0}} \left( \mathfrak{h}_{(n)} \otimes t^{n/6} \right) \oplus \mathbb{C}c \oplus \mathbb{C}d \\ &= \mathfrak{s}_{+} \oplus \mathbb{C}c \oplus \mathbb{C}d \quad (\text{since } \mathfrak{h}_{(0)} = 0). \end{aligned} \tag{2.2.7c}$$

Then $\mathfrak{s}$ is a Heisenberg subalgebra in the sense that its commutator, $[\mathfrak{s}, \mathfrak{s}] = \mathbb{C}c$ is 1-dimensional and coincides with its center.

Consider $\mathbb{C}$ as a 1-dimensional $\mathfrak{b}[\nu]$-module on which $\mathfrak{s}_+$ and $d$ act trivially, and $c$ acts as identity. Form the induced $\widetilde{\mathfrak{h}}[\nu]$-module

$$S = \mathcal{U}\left(\widetilde{\mathfrak{h}}[\nu]\right) \otimes_{\mathcal{U}(\mathfrak{b})} \mathbb{C} \cong \mathcal{S}(\mathfrak{s}_-). \tag{2.2.8}$$

Then $S$ is an irreducible module for the Heisenberg subalgebra $\mathfrak{s}$ (see [FLM88]).

The action of $d$ defines a $\frac{1}{6}\mathbb{Z}$-grading on $S$

$$S = \coprod_{n \in -\frac{1}{6}\mathbb{N}} S_n. \tag{2.2.9}$$

For $\alpha \in \mathfrak{h}$ and $n \in \mathbb{Z}$ define $\alpha_{(n)}$ as the projection of $\alpha$ on to $\mathfrak{h}_{(n)}$. Then $\alpha_{(n)} = 0$ unless $n \equiv \pm 1 \pmod 6$. For $n \in \mathbb{Z}$, define the operator $\alpha(n) = \alpha_{(n)} \otimes t^{n/6}$ on $S$.

**Definition 2.2.1.** Define a pair of Laurent series in $z^{1/6}$ with coefficients in $\mathrm{End}(S)$,

$$\begin{aligned} E^{\pm}(\alpha; z) &= \exp\left( \sum_{\substack{n \in \frac{1}{6}\mathbb{Z} \\ \pm n > 0}} \alpha(6n) \frac{z^{-n}}{n} \right) \\ &= \exp\left( 6 \sum_{\substack{n \in \mathbb{Z} \\ \pm n > 0}} \alpha(n) \frac{z^{-n/6}}{n} \right). \end{aligned} \tag{2.2.10}$$

**Notation 2.2.2.** Fix an $\alpha \in \mathfrak{h}$. For $n \in \mathbb{Z}$, denote by

$$E(n) = E_\alpha(n) = \begin{cases} \text{Coefficient of } z^{-n/6} \text{ in } E^+(\alpha; z) & \text{if } n \geq 0, \\ \text{Coefficient of } z^{-n/6} \text{ in } E^-(\alpha; z) & \text{if } n \leq 0. \end{cases} \tag{2.2.11a}$$

*Remark.* Notice that the constant term in both $E^+(\alpha; z)$ and $E^-(\alpha; z)$ is 1 (the identity operator on $S$). Therefore $E(0) = 1$ is well-defined. The operator $E(n)$ is homogeneous of degree $\frac{n}{6}$. We also have $E(1) = 6\alpha(1)$ and $E(-1) = -6\alpha(-1)$.

Thus,

$$E^+(\alpha; z) = \sum_{\substack{n \in \mathbb{Z} \\ n \geq 0}} E_\alpha(n) z^{-n/6}, \tag{2.2.12a}$$

$$E^-(\alpha; z) = \sum_{\substack{n \in \mathbb{Z} \\ n \geq 0}} E_\alpha(-n) z^{n/6}. \tag{2.2.12b}$$

**Proposition 2.2.3.** *For $\alpha, \beta \in \mathfrak{h}$, we have*

$$E^{+}(\alpha; z_1)E^{-}(\beta; z_2) = E^{-}(\beta; z_2)E^{+}(\alpha; z_1)\prod_{p\in\mathbb{Z}_6}\left(1 - w^{-p}\frac{z_2^{1/6}}{z_1^{1/6}}\right)^{\langle \nu^p\alpha, \beta\rangle}. \tag{2.2.13}$$

*Proof.* Follows from [Proposition 3.4 of LW84, p. 224]. □

The last factor in (2.2.13) is to be expanded as a power series in $\left(\frac{z_2}{z_1}\right)^{1/6}$. We will use the function notation as a short-hand for its power series expansion for brevity. Let

$$\frac{Q_0[\alpha, \beta]}{P_0[\alpha, \beta]} = \prod_{p\in\mathbb{Z}_6}\left(1 - \omega^{-p}\frac{z_2^{1/6}}{z_1^{1/6}}\right)^{\langle \nu^p\alpha, \beta\rangle}, \tag{2.2.14}$$

where $Q_0 = Q_0[\alpha, \beta]$ and $P_0 = P_0[\alpha, \beta]$ are relatively prime polynomials in $\left(\frac{z_2}{z_1}\right)^{1/6}$ with constant term 1, depending on the roots $\alpha, \beta$. We present below a few concrete cases. For $\alpha \in \mathfrak{h}$ with $\langle \alpha, \alpha\rangle = 2$, and letting $x = \left(\frac{z_2}{z_1}\right)^{1/6}$:

$$\begin{aligned}\frac{Q_0[\alpha, \alpha]}{P_0[\alpha, \alpha]} &= \frac{(1-x)^2(1-\omega^{-1}x)(1-\omega^{-5}x)}{(1-\omega^{-2}x)(1-\omega^{-3}x)^2(1-\omega^{-4}x)} \\ &= \frac{(1-x)^6(1-x^6)}{(1-x^2)^3(1-x^3)^2},\end{aligned} \tag{2.2.15a}$$

$$\frac{Q_0[\alpha, \nu\alpha]}{P_0[\alpha, \nu\alpha]} = \frac{(1-x)(1-\omega^{-1}x)^2(1-\omega^{-2}x)}{(1-\omega^{-3}x)(1-\omega^{-4}x)^2(1-\omega^{-5}x)}, \tag{2.2.15b}$$

$$\frac{Q_0[\alpha, \nu^2\alpha]}{P_0[\alpha, \nu^2\alpha]} = \frac{(1-\omega^{-1}x)(1-\omega^{-2}x)^2(1-\omega^{-3}x)}{(1-x)(1-\omega^{-4}x)(1-\omega^{-5}x)}, \tag{2.2.15c}$$

$$\begin{aligned}\frac{Q_0[\alpha, -\alpha]}{P_0[\alpha, -\alpha]} &= \frac{(1-\omega^{-2}x)(1-\omega^{-3}x)^2(1-\omega^{-4}x)}{(1-x)^2(1-\omega^{-1}x)(1-\omega^{-5}x)} \\ &= \frac{(1-x^2)^3(1-x^3)^2}{(1-x)^6(1-x^6)}.\end{aligned} \tag{2.2.15d}$$

The construction of the vertex operator $X(a; z)$ (to be described in what follows), simplifies a lot from the general construction, as described in [FLM88, FLM87, Lep85, Cap92, Cap93], using the following properties of our special case:

(a) $\nu$ is fixed point free, i.e., $\mathfrak{h}_{(0)} = 0$,

(b) $\sum_{p\in\mathbb{Z}_6} \nu^p\alpha = 0$ for $\alpha \in L$,

(c) $\left\langle \sum_{p\in\mathbb{Z}_6} p\nu^p\alpha, \beta\right\rangle \in 6\mathbb{Z}$ for $\alpha, \beta \in L$,

(d) the central extension $\widehat{L}$ (as defined below) splits.

Define the alternate bilinear map $C : L \times L \to \mathbb{C}^*$ by

$$C(\alpha, \beta) = (-1)^{\langle \sum \nu^p \alpha, \beta \rangle} \omega^{\langle \sum p\nu^p \alpha, \beta \rangle}, \tag{2.2.16}$$

where the sums range over $p \in \mathbb{Z}_6$. Note that $C(\alpha, \beta) = 1$ for any $\alpha, \beta \in L$ by (b) and (c) above. There is a unique (up to equivalence) central extension of $L$

$$1 \to \langle \omega \rangle \to \widehat{L} \xrightarrow{-} L \to 0 \tag{2.2.17}$$

by the cyclic group generated by $\omega$ with the commutator map $C$, i.e.,

$$aba^{-1}b^{-1} = C(\bar{a}, \bar{b}) \quad \text{for } a, b \in \widehat{L}. \tag{2.2.18}$$

We use additive notation for the abelian group $L$, and multiplicative notation for the extension $\widehat{L}$ which is not abelian in general.

For $\mathrm{A}_2^{(2)}$, since $C(\alpha, \beta) = 1$ for all $\alpha, \beta \in L$, the above extension splits. Therefore, $\widehat{L} = \langle \omega \rangle \times L$ is the direct product of groups, and is abelian. However, we continue to use the multiplicative notation to be consistent with the notations used for the general case.

Let $\widehat{\nu}$ be the lifting of $\nu$ to $\widehat{L}$ fixing $\omega$, such that

$$\overline{(\widehat{\nu} a)} = \nu \bar{a} \quad \text{for } a \in \widehat{L}, \tag{2.2.19}$$

$$\widehat{\nu} a = a \iff \nu \bar{a} = \bar{a}. \tag{2.2.20}$$

In our case, the extension splits, and therefore $\widehat{\nu}(\omega^p, \alpha) = (\omega^p, \nu\alpha)$ under the identification of $\widehat{L}$ with the direct product $\langle \omega \rangle \times L$.

Let $\widehat{L}$ act on $S = \mathcal{S}(\mathfrak{s}_-)$ as follows:

$$a.s = \omega^p s, \qquad \text{for } a = (\omega^p, \alpha) \in \widehat{L},\ s \in S. \tag{2.2.21}$$

**Definition 2.2.4.** For $a \in \widehat{L}$, define the corresponding vertex operator $X(a; z)$ with coefficients in $\operatorname{End} S$ as follows:

$$X(a; z) = 6^{-\langle \bar{a}, \bar{a} \rangle/2} \sigma(\bar{a}) E^-(-\bar{a}; z) E^+(-\bar{a}; z) a, \tag{2.2.22}$$

where $\sigma(\alpha) = \sigma(\nu\alpha)$ is a normalizing constant depending on $\alpha \in L$, defined by

$$\sigma(\alpha) = 2^{-\langle \alpha, \alpha \rangle/2} (1 - \omega^{-1})^{\langle \nu\alpha, \alpha \rangle} (1 - w^{-2})^{\langle \nu^2 \alpha, \alpha \rangle}. \tag{2.2.23}$$

For $\alpha \in L$ with $\langle \alpha, \alpha \rangle = 2$, the above formula simplifies to

$$\begin{aligned} \sigma(\alpha) &= 2^{-1}(1-\omega^{-1})(1-\omega^{-2})^{-1} \\ &= \frac{\omega_0\sqrt{3}}{6} \end{aligned} \tag{2.2.24}$$

where $\omega_0 = \frac{1}{\sqrt{3}}(1+\omega) = e^{i\pi/6}$ is a 12-th root of unity with $\omega = \omega_0^2$.

Since the these elements $\alpha$ of $L$ play an important role, we shall use the following notations:

$$\begin{aligned} L_2 &= \left\{ \alpha \in L \,\middle|\, \langle \alpha, \alpha \rangle = 2 \right\}, \\ \widehat{L}_2 &= \left\{ a \in \widehat{L} \,\middle|\, \langle \bar{a}, \bar{a} \rangle = 2 \right\}. \end{aligned} \tag{2.2.25}$$

More generally,

$$\begin{aligned} L_n &= \left\{ \alpha \in L \,\middle|\, \langle \alpha, \alpha \rangle = n \right\}, \\ \widehat{L}_n &= \left\{ a \in \widehat{L} \,\middle|\, \langle \bar{a}, \bar{a} \rangle = n \right\}. \end{aligned} \tag{2.2.26}$$

We have the following properties:

$$X(\widehat{\nu} a; z) = \lim_{z^{1/6} \to \omega^{-1} z^{1/6}} X(a; z), \tag{2.2.27}$$

$$\mathrm{D}\, X(a; z) = -[d, X(a; z)], \tag{2.2.28}$$

where $\mathrm{D} = z\frac{\mathrm{d}}{\mathrm{d}z}$.

For $\alpha \in L$, define

$$\begin{aligned} \alpha(z) &= \sum_{n \in \mathbb{Z}} \alpha(n) z^{-\frac{n}{6}-1}, \\ \alpha^+(z) &= \sum_{\substack{n \in \mathbb{Z} \\ n > 0}} \alpha(n) z^{-\frac{n}{6}-1} \\ \alpha^-(z) &= \sum_{\substack{n \in \mathbb{Z} \\ n < 0}} \alpha(n) z^{-\frac{n}{6}-1} \end{aligned} \tag{2.2.29}$$

We will now present the commutator formula for the vertex operators. The details can be found in [Lep85, Cap92, Cap93, FLM87, DL96] in a more general setting, with slightly different notations.

It is sometimes useful to parametrize the vertex operators in terms of the elements of $L$, instead of $\widehat{L}$. This is easy in our case, since the extension $\widehat{L}$ splits. However, the formula becomes more transparent if we follow the more general case. The constants in this formula depend on some normalized sections and normalized cocycles of the extension. For this reason we will describe the general process, and simplify for our case, whenever appropriate.

Let $e : L \to \widehat{L}$ be a normalized section, i.e., $\bar{e}_\alpha = \alpha$. and $e_0 = 1$ for all $\alpha \in L$. Then there is a normalized cocycle $\varepsilon_C : L \times L \to \langle \omega \rangle$ associated with $C$, defined by

$$e_\alpha e_\beta = \varepsilon_C(\alpha, \beta) e_{\alpha+\beta} \quad \text{for } \alpha, \beta \in L, \tag{2.2.30}$$

satisfying

$$\begin{gathered} \varepsilon_C(\alpha, \beta)\varepsilon_C(\alpha+\beta, \gamma) = \varepsilon_C(\beta, \gamma)\varepsilon_C(\alpha, \beta+\gamma), \\ \varepsilon_C(0,0) = 1, \\ \frac{\varepsilon_C(\alpha, \beta)}{\varepsilon_C(\beta, \alpha)} = C(\alpha, \beta). \end{gathered} \tag{2.2.31}$$

In our case, take $e_\alpha = (1, \alpha) \in \widehat{L}$. Therefore, $\varepsilon_C(\alpha, \beta) = 1$ for all $\alpha, \beta \in L$. Also define the map $\eta : \mathbb{Z}_6 \times L \to \langle \omega \rangle$ by

$$\widehat{\nu} e_\alpha = \eta(p, \alpha) e_{\nu^p \alpha}, \tag{2.2.32}$$

which, in our case, simplifies to $\eta(p, \alpha) = 1$ for all $p \in \mathbb{Z}_6$ and $\alpha \in L$.

Define $\varepsilon_2 : L \times L \to \langle \omega \rangle$ by

$$\varepsilon_2(\alpha, \beta) = (-1)^{\langle \sum \nu^p \alpha, \beta \rangle} \omega^{-\langle \sum p\nu^p \alpha, \beta \rangle}, \tag{2.2.33}$$

where the sums range over $-3 < p < 0$. Therefore, the above formula simplifies to

$$\varepsilon_2(\alpha, \beta) = (-1)^{\langle \nu^{-1}\alpha + \nu^{-2}\alpha, \beta \rangle} \omega^{\langle \nu^{-1}\alpha + 2\nu^{-2}\alpha, \beta \rangle}. \tag{2.2.34}$$

This map satisfies

$$\frac{\varepsilon_2(\alpha, \beta)}{\varepsilon_2(\beta, \alpha)} = (-1)^{\langle \alpha, \beta \rangle} C(\alpha, \beta)^{-1}. \tag{2.2.35}$$

Define $\varepsilon : L \times L \to \langle \omega \rangle$ by

$$\varepsilon(\alpha, \beta) = \varepsilon_2(\alpha, \beta)\varepsilon_C(\alpha, \beta), \quad \text{for all } \alpha, \beta \in L. \tag{2.2.36}$$

This is a normalized cocycle associated with the bilinear map $(-1)^{\langle \alpha, \beta \rangle}$. In our case, the map $\varepsilon$ reduces to $\varepsilon_2$.

Using the above notation, we set $X(\alpha; z) = X(e_\alpha; z)$. For $\alpha \in L$, $a = e_\alpha$ we have

$$\begin{aligned} X(\alpha; z) &= 6^{-\langle \alpha, \alpha \rangle/2} \sigma(\alpha) E^-(-\alpha; z) E^+(-\alpha; z), \\ X(a; z) &= 6^{-\langle \bar{a}, \bar{a} \rangle/2} \sigma(\bar{a}) E^-(-\bar{a}; z) E^+(-\bar{a}; z). \end{aligned} \tag{2.2.37}$$

For $\alpha \in L_2$, the constant in front of the above equations simplifies to $\frac{\omega_0\sqrt{3}}{6}$ by (2.2.24).

**Notation 2.2.5.** Write the coefficient of $z^{-n/6}$ in $X(a;z)$ (respectively, in $X(\alpha;z)$) as $X(a;n) \in \operatorname{End} S$, (respectively, $X(\alpha;n) \in \operatorname{End} S$).

$$\begin{aligned} X(a;z) &= \sum_{n\in\mathbb{Z}} X(a;n)z^{-n/6}, \\ X(\alpha;z) &= \sum_{n\in\mathbb{Z}} X(\alpha;n)z^{-n/6}. \end{aligned} \tag{2.2.38}$$

$X(a,n)$ (respectively, $X(\alpha;n)$) is a well-defined operator on $S$ of degree $\frac{n}{6}$.

*Remark.* With this notation, we have

$$X(\nu^k\alpha;n) = \omega^{kn}X(\alpha;n), \tag{2.2.39}$$

for all $n, k \in \mathbb{Z}$.

With the above simplifications the commutator formula of [Lep85] becomes:

**Proposition 2.2.6.** *Let $\alpha, \beta \in L_2$. Set $I(n) = \{p \in \mathbb{Z}_6 \mid \langle \nu^p\alpha, \beta\rangle = n\}$, for $n \in \mathbb{Z}$. Then, setting $x = \left(\frac{z_2}{z_1}\right)^{1/6}$, we have*

$$\begin{aligned} [X(\alpha;z_1), X(\beta;z_2)] = \frac{1}{6} &\sum_{p\in I(-1)} \varepsilon(\nu^p\alpha,\beta)X(\nu^p\alpha+\beta;z_2)\delta\left(\omega^{-p}x\right) \\ &+ \frac{1}{6^2}\varepsilon(-\beta,\beta)\, c \sum_{p\in I(-2)} \mathrm{D}\,\delta\left(\omega^{-p}x\right) \\ &- \frac{1}{6}\varepsilon(-\beta,\beta) \sum_{p\in I(-2)} z_2\beta(z_2)\delta\left(\omega^{-p}x\right), \end{aligned} \tag{2.2.40}$$

*where $c = \mathbf{1} \in \operatorname{End} S$, is the identity endomorphism.*

Because of (2.2.27), and the symmetry of $L_2$ with respect to $\nu$, it is enough to know the commutator $[X(\alpha;z_1), X(\alpha,z_2)]$. We present the formula with the constants simplified below.

**Corollary 2.2.7.** *For $\alpha \in L_2$, we have*

$$\begin{aligned} [X(\alpha;z_1), X(\alpha;z_2)] = &\frac{\omega^2}{6}X(\nu\alpha;z_2)\delta(\omega^{-2}x) - \frac{\omega^2}{6}X(\nu^{-1}\alpha;z_2)\delta(\omega^2x) \\ &+ \frac{\omega}{36}c\,\mathrm{D}\,\delta(-x) - \frac{\omega}{6}z_2\alpha(z_2)\delta(-x), \end{aligned} \tag{2.2.41}$$

*where $x = \left(\frac{z_2}{z_1}\right)^{1/6}$.*

*Proof.* Use

$$I(-1) = \left\{ p \in \mathbb{Z}_6 \;\middle|\; \langle \nu^p \alpha, \alpha \rangle = -1 \right\} = \{\pm 2\},$$
$$I(-2) = \left\{ p \in \mathbb{Z}_6 \;\middle|\; \langle \nu^p \alpha, \alpha \rangle = -2 \right\} = \{3\},$$
$$\nu^2\alpha + \alpha = \nu\alpha,$$
$$\nu^{-2}\alpha + \alpha = \nu^{-1}\alpha,$$
$$\varepsilon(\nu^2\alpha, \alpha) = \omega^2,$$
$$\varepsilon(\nu^{-2}\alpha, \alpha) = -\omega^2,$$
$$\varepsilon(-\alpha, \alpha) = \omega$$

in the above Proposition 2.2.6. □

We need a few more commutator relations. Recall the "delta function" Laurent series as defined in (2.1.2). Define the following related Laurent series:

$$\delta^+(z) = \sum_{n>0} z^n, \tag{2.2.42a}$$

$$\delta^-(z) = \sum_{n>0} z^{-n}, \tag{2.2.42b}$$

$$\delta_{\{\pm 1\}}(z) = \sum_{\substack{n \equiv \pm 1 \\ (\text{mod } 6)}} z^n, \tag{2.2.42c}$$

$$\delta^+_{\{\pm 1\}}(z) = \sum_{\substack{n>0 \\ n \equiv \pm 1 \pmod 6}} z^n, \tag{2.2.42d}$$

$$\delta^-_{\{\pm 1\}}(z) = \sum_{\substack{n>0 \\ n \equiv \pm 1 \pmod 6}} z^{-n}. \tag{2.2.42e}$$

Also recall the operator $\mathrm{D}\colon z^n \mapsto nz^n$. Define the inverse operator

$$\mathrm{D}^{-1}\colon z^n \mapsto \frac{z^n}{n}, \tag{2.2.43}$$

for $n \neq 0$.

Recall the operator on $S$ given by $\alpha(n) = \alpha_{(n)} \otimes t^{n/6}$ for $n \in \mathbb{Z}$, and the Laurent series $\alpha(z)$, $\alpha^+(z)$ and $\alpha^-(z)$ defined in (2.2.29). Note that $\alpha(n)$ is the coefficient of $z^{-n/6}$ in $z\alpha(z)$.

With these notations, we have

$$\begin{aligned} E^+(-\alpha; z) &= \exp\left(\mathrm{D}^{-1}\, z\alpha^+(z)\right), \\ E^-(-\alpha; z) &= \exp\left(\mathrm{D}^{-1}\, z\alpha^-(z)\right). \end{aligned} \tag{2.2.44}$$

**Proposition 2.2.8.** *For $\alpha \in \mathfrak{h}$ we have*

$$[z_1\alpha^-(z_1), E^+(-\alpha; z_2)] = E^+(-\alpha; z_2)\, \delta^-_{\{\pm 1\}}(x), \tag{2.2.45a}$$

$$[z_1\alpha^+(z_1), E^-(-\alpha; z_2)] = E^-(-\alpha; z_2)\, \delta^+_{\{\pm 1\}}(x), \tag{2.2.45b}$$

*where $x = \left(\frac{z_2}{z_1}\right)^{1/6}$.*

*Proof.* Let $\mathrm{D}_2^{-1}$ denote the operator as defined in (2.2.43) operating on the variable $z_2$. Then

$$\begin{aligned} [z_1\alpha^-(z_1), E^+(-\alpha; z_2)] &= \left[z_1\alpha^-(z_1), \exp\left(\mathrm{D}_2^{-1}\, z_2\alpha^+(z_2)\right)\right] \\ &= \exp\left(\mathrm{D}_2^{-1}\, z_2\alpha^+(z_2)\right) \cdot [z_1\alpha^-(z_1), \mathrm{D}_2^{-1}\, z_2\alpha^+(z_2)] \\ &= \exp\left(\mathrm{D}_2^{-1}\, z_2\alpha^+(z_2)\right) \cdot \mathrm{D}_2^{-1}[z_1\alpha^-(z_1), z_2\alpha^+(z_2)] \\ &= \exp\left(\mathrm{D}_2^{-1}\, z_2\alpha^+(z_2)\right) \cdot \mathrm{D}_2^{-1} \sum_{n>0} [\alpha(-n), \alpha(n)] z_1^{n/6} z_2^{-n/6} \\ &= \exp\left(\mathrm{D}_2^{-1}\, z_2\alpha^+(z_2)\right) \cdot \mathrm{D}_2^{-1} \sum_{\substack{n>0 \\ n \equiv \pm 1 \pmod 6}} \left(-\frac{n}{6}\right) z_1^{n/6} z_2^{-n/6} \\ &= \exp\left(\mathrm{D}_2^{-1}\, z_2\alpha^+(z_2)\right) \cdot \sum_{\substack{n>0 \\ n \equiv \pm 1 \pmod 6}} z_1^{n/6} z_2^{-n/6} \\ &= E^+(-\alpha; z_2)\, \delta^-_{\{\pm 1\}}(x). \end{aligned}$$

The second equation follows from similar calculation. $\square$

**Corollary 2.2.9.** *For $\alpha \in L_2$,*

$$[z_1\alpha^-(z_1), X(\alpha; z_2)] = X(\alpha; z_2)\, \delta^-_{\{\pm 1\}}(x), \tag{2.2.46a}$$

$$[z_1\alpha^+(z_1), X(\alpha; z_2)] = X(\alpha; z_2)\, \delta^+_{\{\pm 1\}}(x), \tag{2.2.46b}$$

*where $x = \left(\frac{z_2}{z_1}\right)^{1/6}$.*

*Proof.* Follows from the Proposition 2.2.8, the definition (2.2.37) of $X(\alpha; z)$ and the fact that

$$[z_1\alpha^-(z_1), E^-(-\alpha; z_2)] = 0,$$
$$[z_1\alpha^+(z_1), E^+(-\alpha; z_2)] = 0. \qquad \square$$

**Corollary 2.2.10.** *For $\alpha \in L_2$,*

$$[z_1\alpha(z_1), X(\alpha; z_2)] = X(\alpha; z_2)\delta_{\{\pm 1\}}(x),$$

*where $x = \left(\frac{z_2}{z_1}\right)^{1/6}$.*

*Proof.* Follows from Corollary 2.2.9 and the fact that $z\alpha(z) = z\alpha^-(z) + z\alpha^+(z)$. $\square$

For completeness, we give the following formula which will be useful later.

**Proposition 2.2.11.** *Let $\alpha \in L_2$. Then we have*

$$E^+(-\alpha; z_1)X(\alpha; z_2) = \Psi(x)X(\alpha; z_2)E^+(-\alpha; z_1), \tag{2.2.47a}$$

$$X(\alpha; z_1)E^-(-\alpha; z_2) = \Psi(x)E^-(-\alpha; z_2)X(\alpha; z_1), \tag{2.2.47b}$$

*where $\Psi(x) = \frac{Q_0[-\alpha,-\alpha]}{P_0[-\alpha,-\alpha]} = \frac{Q_0[\alpha,\alpha]}{P_0[\alpha,\alpha]}$ is a power series, and $P_0$, $Q_0$ are polynomials in $x = \left(\frac{z_2}{z_1}\right)^{1/6}$ as defined in* (2.2.14) *and* (2.2.15a).

*Proof.* Follows from the definition of $X(\alpha; z)$ (2.2.37) and Proposition 2.2.3. $\square$

The following corollary of the above proposition is particularly important for the later discourse.

**Corollary 2.2.12.** *Let $X(n) = X(\alpha; n)$, $E'(n) = E(-\alpha; n)$ and $\Psi(n)$ be the coefficient of $x^n = \left(\frac{z_2}{z_1}\right)^{n/6}$ in $\Psi(x)$. Then*

$$X(-m)E'(-n) = \sum_{k=0}^{n} \Psi(k)E'(-(n-k))X(-(m+k)), \tag{2.2.48}$$

*for $m, n \in Z$, $n \geq 0$.*

*Proof.* The result follows from (2.2.47b), by comparing the coefficient of $z_1^{m/6} z_2^{n/6}$ on both sides, and the fact that $E^-(-\alpha; z_2)$ only has the nonnegative powers of $z_2^{1/6}$. $\square$

Since $\Psi(x) = \frac{Q_0[\alpha,\alpha]}{P_0[\alpha,\alpha]}$ plays a very important role, we give the first few terms of $\Psi(x)$.

$$\Psi(x) = 1 - 6x + 18x^2 - 36x^3 + 54x^4 - 66x^5 + 72x^6 - 78x^7 + \dots. \tag{2.2.49}$$

We present the following commutation relations which will be needed later to show explicitly the isomorphism of the algebra $\mathrm{A}_2^{(2)}$ with the vertex operator representation.

**Proposition 2.2.13.** *Fix any $\alpha \in L_2$. Let $X(n) = X(\alpha; n)$. Then we have, for $m, n \in \mathbb{Z}$,*

$$[\alpha(m), \alpha(n)] = \frac{m}{6}\,\delta_{m+n,0}\,c, \qquad \textit{if } m, n \equiv \pm 1 \pmod 6 \tag{2.2.50a}$$

$$[\alpha(m), X(n)] = X(m+n), \qquad \textit{if } m \equiv \pm 1 \pmod 6, \tag{2.2.50b}$$

$$\begin{aligned}[X(m), X(n)] = {} & \frac{\omega^2}{6}\left(\omega^{n-m} - \omega^{m-n}\right) X(m+n) \\ & - \frac{\omega}{6}(-1)^m \alpha(m+n) \\ & + \delta_{m+n,0}\,\frac{\omega}{36}(-1)^m mc.\end{aligned} \tag{2.2.50c}$$

*In particular,*

$$[X(0), X(1)] = -\frac{\omega_0\sqrt{3}}{6}X(1) - \frac{\omega}{6}\alpha(1), \tag{2.2.50d}$$

$$[X(0), X(-1)] = \frac{\omega_0\sqrt{3}}{6}X(-1) - \frac{\omega}{6}\alpha(-1), \tag{2.2.50e}$$

$$[X(1), X(-1)] = \frac{\omega_0\sqrt{3}}{6}X(0) - \frac{\omega}{36}c, \tag{2.2.50f}$$

*where $\omega = e^{\pi i/3}$ and $\omega_0 = e^{\pi i/6}$ (primitive 6th and 12th roots of unity respectively, such that $\omega_0^2 = \omega$).*

*Proof.* The first equation (2.2.50a) follows from (2.2.6). Equation (2.2.50b) follows from Corollary 2.2.10 by equating the coefficients of $z_1^{-m/6} z_2^{-n/6}$ on both sides. Equation (2.2.50c) follows from Corollary 2.2.7 by equating the coefficients of $z_1^{-m/6} z_2^{-n/6}$ on both sides.

The next three special cases follows from (2.2.50c) with the simplification

$$\omega^{n-m} - \omega^{m-n} = \begin{cases} 0 & \text{if } n - m \equiv 0, 3 \pmod 6 \\ \sqrt{3}i & \text{if } n - m \equiv 1, 2 \pmod 6 \\ -\sqrt{3}i & \text{if } n - m \equiv 4, 5 \pmod 6 \end{cases} \tag{2.2.51}$$

(using $i = \omega_0^3$ and $\omega = \omega_0^2$). $\square$

## 2.3 The Algebra $\mathrm{A}_2^{(2)}$

In this section, we present a brief description of the affine Lie algebra of the type $\mathrm{A}_2^{(2)}$ in terms of the generators and relations (see [Kac90] for more details). Then we describe the principal realization of this algebra using vertex operator representation on $S$ (see [FLM88, Lep85]), with the explicit image of the generators under this isomorphism.

The algebra $\widehat{\mathfrak{g}}$ of the type $\mathrm{A}_2^{(2)}$ is the Kac-Moody algebra associated with the generalized Cartan matrix

$$A = \begin{pmatrix} 2 & -4 \\ -1 & 2 \end{pmatrix}, \tag{2.3.1}$$

with the Lie algebra generators $h_0, h_1, e_0, e_1, f_0, f_1$ and the relations

$$\begin{aligned} [h_i, h_j] &= 0, \\ [h_i, e_j] &= a_{ij} e_j, \\ [h_i, f_j] &= -a_{ij} f_j, \\ [e_i, f_j] &= \delta_{ij} h_i, \\ (\operatorname{ad} e_i)^{-a_{ij}+1} e_j &= 0 \quad (\text{for } i \neq j), \\ (\operatorname{ad} f_i)^{-a_{ij}+1} f_j &= 0 \quad (\text{for } i \neq j), \end{aligned} \tag{2.3.2}$$

for all $i, j \in \{0, 1\}$, where $a_{ij}$ denotes the $(i, j)$-th entry of the above generalized Cartan matrix $A$ (indexed from 0). It follows from the relations that $c = h_0 + 2h_1$ is central.

The principal $\frac{1}{6}\mathbb{Z}$-gradation of $\widehat{\mathfrak{g}}$ is given by assigning

$$\begin{aligned} \deg h_i &= 0, \\ \deg e_i &= 1/6, \\ \deg f_i &= -1/6, \end{aligned} \tag{2.3.3}$$

for all $i \in \{0, 1\}$.

It is sometimes useful to work with the extended algebra $\widetilde{\mathfrak{g}}$ of type $\mathrm{A}_2^{(2)}$. $\widetilde{\mathfrak{g}}$ is the extension of $\widehat{\mathfrak{g}}$ by a degree derivation $d$:

$$\widetilde{\mathfrak{g}} = \widehat{\mathfrak{g}} \oplus \mathbb{C}d, \tag{2.3.4}$$

with the brackets

$$[d, x] = (\deg x)\, x, \tag{2.3.5}$$

for any $x \in \widehat{\mathfrak{g}}$ of homogeneous degree. Note that $\widehat{\mathfrak{g}} = [\widetilde{\mathfrak{g}}, \widetilde{\mathfrak{g}}]$ is the commutator subalgebra of $\widetilde{\mathfrak{g}}$. The advantage of working with the the extended algebra $\widetilde{\mathfrak{g}}$ is that the gradation becomes intrinsic.

Now we describe the principal realization of $\mathrm{A}_2^{(2)}$ using the vertex operators defined in the previous section (§ 2.2). Although, this construction as described in [FLM88, Lep85] is quite deep and complicated, it is a lot simpler for the case of $\mathrm{A}_2^{(2)}$ because of the simplification that happens in this particular case. We give a very brief description, simplified for this particular case.

Recall that in § 2.2, we started with the even lattice $L$ of type $\mathrm{A}_2$. Define a Lie algebra $\mathfrak{g}$ as the direct sum

$$\mathfrak{g} = \mathfrak{h} \bigoplus_{\alpha \in L_2} \mathbb{C} x_\alpha, \tag{2.3.6}$$

with the brackets $[\mathfrak{h}, \mathfrak{h}] = 0$, $[h, x_\alpha] = \langle h, \alpha \rangle x_\alpha$, and

$$[\mathfrak{h}, \mathfrak{h}] = 0, \tag{2.3.7a}$$

$$[h, x_\alpha] = \langle h, \alpha \rangle x_\alpha, \tag{2.3.7b}$$

$$[x_\alpha, x_\beta] = \begin{cases} \epsilon(\alpha, -\alpha)\alpha & \text{if } \alpha + \beta = 0 \\ \epsilon(\alpha, \beta) x_{\alpha+\beta} & \text{if } \langle \alpha, \beta \rangle = -1 \\ 0 & \text{if } \langle \alpha, \beta \rangle \geq 0, \end{cases} \tag{2.3.7c}$$

for $h \in \mathfrak{h}$ and $\alpha, \beta \in L_2$. Then $\mathfrak{g}$ is a Lie algebra (see [FK80, Seg81, FLM88]). Extend the automorphism $\nu$ of $\mathfrak{h}$ to $\mathfrak{g}$ by (recall $\eta(p, \alpha)$ from (2.2.32)):

$$\begin{aligned} \nu x_\alpha &= \eta(1, \alpha) x_{\nu\alpha} \\ &= x_{\nu\alpha} \qquad \text{(since } \eta \equiv 1 \text{ in this case)}, \end{aligned} \tag{2.3.8}$$

and the form $\langle\, , \,\rangle$ by

$$\langle h, x_\alpha \rangle = 0, \tag{2.3.9a}$$

$$\langle x_\alpha, x_\beta \rangle = \begin{cases} \epsilon(\alpha, -\alpha) & \text{if } \alpha + \beta = 0 \\ 0 & \text{if } \alpha + \beta \neq 0 \end{cases} \tag{2.3.9b}$$

for $h \in \mathfrak{h}$ and $\alpha, \beta \in L_2$. Then $\langle\, , \,\rangle$ is $\mathfrak{g}$-invariant and preserved by $\nu$. Furthermore, $\mathfrak{g}$ is semisimple ($g \cong \mathfrak{sl}_3$), since $L_2$ spans $\mathfrak{h}$.

Note that $\nu^6 = 1$. Let

$$\mathfrak{g}_{(n)} = \left\{ x \in \mathfrak{g} \,\middle|\, \nu x = \omega^n x \right\} \tag{2.3.10}$$

denote, if nontrivial, the eigenspace for the eigenvalue $\omega^n$, $n \in \mathbb{Z}$. Form the $\nu$-twisted affine Lie algebra

$$\widetilde{\mathfrak{g}}[\nu] = \coprod_{n \in \mathbb{Z}} \mathfrak{g}_{(n)} \otimes t^{n/6} \oplus \mathbb{C}c \oplus \mathbb{C}d \tag{2.3.11}$$

with the brackets

$$\begin{aligned} [x \oplus t^{i/6}, y \oplus t^{j/6}] &= [x, y] \otimes t^{(i+j)/6} + \tfrac{i}{6} \langle x, y \rangle \delta_{i+j,0}\, c, \\ [d, x \oplus t^{i/6}] &= \tfrac{i}{6} x \oplus t^{i/6}, \\ [c, d] &= [c, x \oplus t^{i/6}] = 0, \end{aligned} \tag{2.3.12}$$

where $i, j \in \mathbb{Z}$, $x \in \mathfrak{g}_{(i)}$, $y \in \mathfrak{g}_{(j)}$.

Define

$$x(\alpha; z) = \sum_{n \in \mathbb{Z}} \left( (x_\alpha)_{(n)} \otimes t^{n/6} \right) z^{-n/6}, \tag{2.3.13}$$

where $(x_\alpha)_{(n)}$ denotes the projection of $x_\alpha$ onto $\mathfrak{g}_{(n)}$.

The operators $E_\alpha(n), X(\alpha; n) \in \operatorname{End} S$, for $\alpha \in L_2$ and $n \in \mathbb{Z}$, define a representation of $\widetilde{\mathfrak{h}}[\nu]$ on $S$. By Theorem 9.1 of [Lep85] this representation of $\widetilde{\mathfrak{h}}[\nu]$ on $S$ extends uniquely to an irreducible Lie algebra representation of $\widetilde{\mathfrak{g}}[\nu]$ on $S$ such that

$$x(\alpha; z) \mapsto X(\alpha; z)$$

for all $\alpha \in L_2$.

The Lie algebra $\widetilde{\mathfrak{g}}[\nu]$ can be shown to be isomorphic to the principal ($\frac{1}{6}\mathbb{Z}$-graded) realization of the affine Lie algebra $\widetilde{\mathfrak{g}}$ of type $\mathrm{A}_2^{(2)}$ (see [Fig87, Kac90]). Here, we will show this fact directly using the generators and relations of $\mathrm{A}_2^{(2)}$.

**Proposition 2.3.1.** *Fix any $\alpha \in L_2$. Let $X(n) = X(\alpha; n)$, for $n \in \mathbb{Z}$. The following map establishes the representation of $\widetilde{\mathfrak{g}}[\nu]$ on $S$ as the principal ($\frac{1}{6}\mathbb{Z}$-graded) realization of $\mathrm{A}_2^{(2)}$*

$$\begin{aligned}
h_0 &\mapsto \tfrac{4\sqrt{3}}{\omega_0} X(0) + \tfrac{2}{3} c, \\
h_1 &\mapsto -\tfrac{2\sqrt{3}}{\omega_0} X(0) + \tfrac{1}{6} c, \\
e_0 &\mapsto -\tfrac{2\sqrt{2}}{\omega_0} X(1) + \tfrac{2\sqrt{2}}{\sqrt{3}} \alpha(1), \\
e_1 &\mapsto \tfrac{2}{\omega_0} X(1) + \tfrac{1}{\sqrt{3}} \alpha(1), \\
f_0 &\mapsto \tfrac{2\sqrt{2}}{\omega_0} X(-1) + \tfrac{2\sqrt{2}}{\sqrt{3}} \alpha(-1), \\
f_1 &\mapsto -\tfrac{2}{\omega_0} X(-1) + \tfrac{1}{\sqrt{3}} \alpha(-1), \\
d &\mapsto d,
\end{aligned} \tag{2.3.14}$$

*where $\omega_0 = \sqrt{\omega} = e^{i\pi/6}$, and $c$ is the identity operator on $S$.*

*Proof.* The defining relations can be directly verified using the commutation relations in Proposition 2.2.13. (Also see § B.1). □

*Remark.* Note that $X(n), \alpha(n) \in \widetilde{\mathfrak{g}}[\nu]$. However, the operators $E(n)$, in general, need not be in the Lie algebra, but in the universal enveloping algebra, $\mathcal{U}(\widetilde{\mathfrak{g}}[\nu])$.

We end this section with the definition of three subalgebras of $\widetilde{\mathfrak{g}} = \widetilde{\mathfrak{g}}[\nu]$.

$$\widetilde{\mathfrak{h}}_0 = \operatorname{span} \left\{ x \in \widetilde{\mathfrak{g}} \;\middle|\; \deg x = 0 \right\}, \tag{2.3.15a}$$

$$\mathfrak{n}_\pm = \operatorname{span} \left\{ x \in \widetilde{\mathfrak{g}} \;\middle|\; \pm (\deg x) > 0 \right\}. \tag{2.3.15b}$$

Then

$$\widetilde{\mathfrak{g}} = \mathfrak{n}_- \oplus \widetilde{\mathfrak{h}}_0 \oplus \mathfrak{n}_+. \tag{2.3.16}$$

Note that $\widetilde{\mathfrak{h}}_0$ is the subalgebra spanned by $X(0)$, $c$ and $d$ ($c$ is the identity operator on $S$). The subalgebra $\mathfrak{n}_+$ (respectively, $\mathfrak{n}_-$) is spanned by $X(n)$, $\alpha(n)$ for $n > 0$ (respectively, $n < 0$). In terms of the Chevalley generators, $\widetilde{\mathfrak{h}}_0$ is the subalgebra generated by $h_0$, $h_1$ and $d$; and $\mathfrak{n}_+$ (respectively, $\mathfrak{n}_-$) is the subalgebra generated by $e_0$ and $e_1$ (respectively, $f_0$ and $f_1$).

# Chapter 3

# Standard Modules

The main objects of our study are the level 4 standard modules for the algebra $\widetilde{\mathfrak{g}} = \widetilde{\mathfrak{g}}[\nu]$ of type $\mathrm{A}_2^{(2)}$.

In § 3.1 we recall the basic notions and terminology from the general representation theory of Kac-Moody algebras. We also show that the standard modules of level 4 can be thought of as submodules of the tensor product of four copies of the "basic module." In § 3.2 we present the graded dimensions for the three standard modules of level 4. See [Kac90], [Lep78] for more details.

## 3.1 Basic Notions

Recall the subalgebras $\widetilde{\mathfrak{h}}_0$, $\mathfrak{n}_-$ and $\mathfrak{n}_+$ of zero, negative and positive degree respectively in $\widetilde{\mathfrak{g}}$, as defined in (2.3.15).

Let $V$ be a $\widetilde{\mathfrak{g}}$-module and $\Lambda \in (\widetilde{\mathfrak{h}}_0)^*$. Assume that $\Lambda(d) = 0$. $V$ is called a *highest weight module* with *highest weight* $\Lambda$, if it is generated by an element $v_0 \neq 0$ (called a *highest weight vector*) such that

(i) $\mathfrak{n}_+ \cdot v_0 = 0$,

(ii) $h \cdot v_0 = \Lambda(h) v_0$ for all $h \in \widetilde{\mathfrak{h}}_0$.

The highest weight vector $v_0$ is unique up to multiplication by a nonzero scalar.

An element $\Lambda \in (\widetilde{\mathfrak{h}}_0)^*$ is called *dominant integral* if $\Lambda \neq 0$ and $\Lambda(h_i) \in \mathbb{Z}_{\geq 0}$ for $i = 0, 1$, where $h_i$ are the elements as described in § 2.3.

$V$ is called a *standard* module if

(i) it is a highest weight module with highest weight $\Lambda$,

(ii) it is irreducible,

(iii) $\Lambda$ is dominant integral.

Given a dominant integral weight $\Lambda$, there is a unique standard module with highest weight $\Lambda$, up to equivalence.

**Notation 3.1.1.** For a dominant integral weight $\Lambda$, denote by $L(\Lambda)$ the standard module with highest weight $\Lambda$.

Recall the Chevalley generators $h_0, h_1, e_0, e_1, f_0, f_1$, as described in § 2.3. Define the elements $h_0^*$, $h_1^* \in (\widetilde{\mathfrak{h}}_0)^*$:

$$h_i^*(h_j) = \delta_{ij}, \quad h_i^*(d) = 0, \quad \text{for } i, j \in \{0, 1\}. \tag{3.1.1}$$

Recall that we have $c = h_0 + 2h_1$. Therefore, $\Lambda(c) \in \mathbb{N}$, if $\Lambda$ is dominant integral. *Level* of the standard module $L(\Lambda)$ is the positive integer $\Lambda(c) = \Lambda(h_0) + 2\Lambda(h_1)$.

There is only one, up to equivalence, level 1 standard module called the *basic module*. It is the standard module of highest weight $h_0^*$.

**Notation 3.1.2.** We denote the basic module of $\mathrm{A}_2^{(2)}$ by $U = L(h_0^*)$.

Let $V$ be a standard module of highest weight $\Lambda$ with a highest weight vector $v_0$. Let $k_i = \Lambda(h_i)$, $i = 0, 1$. Then

$$f_0^{k_0+1} v_0 = f_1^{k_1+1} v_0 = 0. \tag{3.1.2}$$

Consider $\mathbb{C}$ as a one-dimensional representation of $\widetilde{\mathfrak{h}}_0 \oplus \mathfrak{n}_+$, such that $h_i \cdot 1 = \Lambda(h_i)$ for $i = 0, 1$ and $\mathfrak{n}_+ \cdot 1 = 0$. Denote by

$$M(\Lambda) = \mathcal{U}\left(\tilde{\mathfrak{g}}[\nu]\right) \otimes_{\mathcal{U}\left(\tilde{\mathfrak{h}}_0 \oplus \mathfrak{n}_+\right)} \mathbb{C}$$

the Verma module with highest weight $\Lambda$. As a vector space

$$M(\Lambda) \cong \mathcal{U}(\mathfrak{n}_-).$$

Then, $L(\Lambda) \cong M(\Lambda)/W(\Lambda)$, where

$$W(\Lambda) = \mathcal{U}(\mathfrak{n}_-) f_0^{k_0+1} v_0 + \mathcal{U}(\mathfrak{n}_-) f_1^{k_1+1} v_0.$$

There are three level 4 standard modules (up to equivalence) for $\mathrm{A}_2^{(2)}$. They are $L(4h_0^*)$, $L(2h_0^* + h_1^*)$ and $L(2h_1^*)$. Now we will show that the level 4 standard $\mathrm{A}_2^{(2)}$-modules are contained in the tensor product of four copies of the basic module $U$. First, we need a couple of lemmas. The first one records various straightening relations in $A_2^{(2)}$.

**Lemma 3.1.3.** *Let $U$ be a basic module for $A_2^{(2)}$ with a highest weight vector $u_0 \in U$. Then we have*

$$\begin{aligned} h_0 u_0 &= u_0, \quad h_1 u_0 = 0, \\ e_0 u_0 &= 0, \quad\ \ e_1 u_0 = 0, \\ f_0^2 u_0 &= 0, \quad\ \ f_1 u_0 = 0. \end{aligned} \tag{3.1.3}$$

*and*

$$\begin{aligned} h_0 f_0 u_0 &= f_0 h_0 u_0 - 2 f_0 u_0 = -f_0 u_0, \\ h_1 f_0 u_0 &= f_0 h_1 u_0 + f_0 u_0 = f_0 u_0, \\ e_0 f_0 u_0 &= f_0 e_0 u_0 + h_0 u_0 = u_0, \\ e_1 f_0 u_0 &= f_0 e_1 u_0 = 0, \\ h_0 f_1 u_0 &= f_1 h_0 u_0 + 4 f_0 u_0 = 5 f_1 u_0, \\ h_1 f_1 u_0 &= f_1 h_1 u_0 - 2 f_1 u_0 = -2 f_1 u_0, \\ e_0 f_1 u_0 &= f_1 e_0 u_0 = 0, \\ e_1 f_1 u_0 &= f_1 e_1 u_0 + h_1 u_0 = 0. \end{aligned} \tag{3.1.4}$$

*Proof.* This follows from the Serre relations (2.3.2) among the Chevalley generators of $A_2^{(2)}$, together with (3.1.2) and the definition of highest weight. □

**Lemma 3.1.4.** *The submodule of $U \otimes U$ generated by $v_0 = f_0 u_0 \otimes u_0 - u_0 \otimes f_0 u_0$ is isomorphic to $L(h_1^*)$, where $u_0$ is a highest weight vector of $U$.*

*Proof.* We need to show that $h_0 v_0 = 0$, $h_1 v_0 = v_0$, $e_0 v_0 = e_1 v_0 = 0$.

Using Lemma 3.1.3, we have

$$\begin{aligned} h_0(f_0 u_0 \otimes u_0) &= h_0 f_0 u_0 \otimes u_0 + f_0 u_0 \otimes h_0 u_0 \\ &= -f_0 u_0 \otimes u_0 + f_0 u_0 \otimes u_0 \\ &= 0, \end{aligned}$$

and similarly, $h_0(u_0 \otimes f_0 u_0) = 0$. Thus, $h_0 v_0 = 0$.

Now,

$$h_1(f_0u_0 \otimes u_0) = h_1f_0u_0 \otimes u_0 + f_0u_0 \otimes h_1u_0$$
$$= f_0u_0 \otimes u_0,$$

and similarly, $h_1(u_0 \otimes f_0u_0) = u_0 \otimes f_0u_0$. Therefore, $h_1v_0 = v_0$.

Using Lemma 3.1.3

$$e_0(f_0u_0 \otimes u_0) = e_0f_0u_0 \otimes u_0 = u_0 \otimes u_0,$$

and similarly, $e_0(u_0 \otimes f_0u_0) = u_0 \otimes u_0$. Therefore, $e_0v_0 = 0$.

And finally,

$$e_1(f_0u_0 \otimes u_0) = e_1f_0u_0 \otimes u_0 = 0.$$

Thus, $e_1v_0 = 0$.

Therefore, we have proved that $v_0$ is a highest weight vector for $h_1^*$. Then the submodule of $U \otimes U$ generated by $v_0$ is isomorphic to the standard module $L(h_1^*)$. □

**Notation 3.1.5.** We use the following notation for brevity:

$$U^{\otimes 4} = U \otimes U \otimes U \otimes U.$$

**Proposition 3.1.6.** *Let $U$ be the basic module for $\mathrm{A}_2^{(2)}$ with a highest weight vector $u_0$. Let $v_0 = f_0u_0 \otimes u_0 - u_0 \otimes f_0u_0 \in U \otimes U$, as defined in Lemma 3.1.4.*

(i) *The submodule of $U^{\otimes 4}$ generated by $u_0 \otimes u_0 \otimes u_0 \otimes u_0$ is isomorphic to $L(4h_0^*)$.*

(ii) *The submodule of $U^{\otimes 4}$ generated by $v_0 \otimes u_0 \otimes u_0$ is isomorphic to $L(2h_0^* + h_1^*)$.*

(iii) *The submodule of $U^{\otimes 4}$ generated by $v_0 \otimes v_0$ is isomorphic to $L(2h_1^*)$.*

*Proof.* Let $V = U \otimes U \otimes U \otimes U$. Fix a highest weight vector $u_0 \in U$.

(i) Let $v = u_0 \otimes u_0 \otimes u_0 \otimes u_0$. Then, we have

$$\begin{aligned} h_0(v) &= h_0u_0 \otimes u_0 \otimes u_0 \otimes u_0 + u_0 \otimes h_0u_0 \otimes u_0 \otimes u_0 \\ &\quad + u_0 \otimes u_0 \otimes h_0u_0 \otimes u_0 + u_0 \otimes u_0 \otimes u_0 \otimes h_0u_0 \\ &= 4v, \end{aligned}$$

by (3.1.3) and (3.1.3). Clearly, $h_1 v = e_0 v = e_1 v = 0$. Thus, $v$ is a highest weight vector for $4h_0^*$. Therefore, the submodule of $V$ generated by $v$ is isomorphic to the standard module $L(4h_0^*)$.

(ii) Let $v = v_0 \otimes u_0 \otimes u_0$. Using Lemma 3.1.3 and Lemma 3.1.4, we have

$$\begin{aligned}
h_0 v &= h_0 v_0 \otimes u_0 \otimes u_0 + v_0 \otimes h_0 u_0 \otimes u_0 + v_0 \otimes u_0 \otimes h_0 u_0 \\
&= 2v_0 \otimes u_0 \otimes u_0 \\
&= 2v, \\
h_1 v &= h_1 v_0 \otimes u_0 \otimes u_0 + v_0 \otimes h_1 u_0 \otimes u_0 + v_0 \otimes u_0 \otimes h_1 u_0 \\
&= v_0 \otimes u_0 \otimes u_0 \\
&= v, \\
e_0 v &= e_0 v_0 \otimes u_0 \otimes u_0 + v_0 \otimes e_0 u_0 \otimes u_0 + v_0 \otimes u_0 \otimes e_0 u_0 \\
&= 0, \\
e_1 v &= e_1 v_0 \otimes u_0 \otimes u_0 + v_0 \otimes e_1 u_0 \otimes u_0 + v_0 \otimes u_0 \otimes e_1 u_0 \\
&= 0
\end{aligned}$$

Thus, $v$ is a highest weight vector for $2h_0^* + h_1^*$. Therefore, the submodule of $V$ generated by $v$ is isomorphic to the standard module $L(2h_0^* + h_1^*)$.

(iii) Let $v = v_0 \otimes v_0$. Using Lemma 3.1.3 and Lemma 3.1.4, we have

$$\begin{aligned}
h_0 v &= h_0 v_0 \otimes v_0 + v_0 \otimes h_0 v_0 \\
&= 0, \\
h_1 v &= h_1 v_0 \otimes v_0 + v_0 \otimes h_1 v_0 \\
&= 2v_0 \otimes v_0 \\
&= 2v, \\
e_0 v &= e_0 v_0 \otimes v_0 + v_0 \otimes e_0 v_0 \\
&= 0, \\
e_1 v &= e_1 v_0 \otimes v_0 + v_0 \otimes e_1 v_0 \\
&= 0.
\end{aligned}$$

Thus, $v$ is a highest weight vector of weight $2h_1^*$. Therefore, the submodule of $V$ generated by $v$ is isomorphic to the standard module $L(2h_1^*)$. $\square$

## 3.2 The Graded Dimensions

In this section, we present the graded dimension formula, obtained from the principal specialization of the Weyl-Kac character formula and the numerator formula of [LM78, Lep78], for each of the three standard $\mathrm{A}_2^{(2)}$-modules of level 4.

We adopt the following alternative notations for these modules.

$$(4,0)\text{-module} = L(4h_0^*), \tag{3.2.1a}$$

$$(2,1)\text{-module} = L(2h_0^* + h_1^*), \tag{3.2.1b}$$

$$(0,2)\text{-module} = L(2h_1^*). \tag{3.2.1c}$$

Let $V$ be any standard module of level $k$ with highest weight $\Lambda$. The structure of $V$ is the same under the actions of $\widetilde{\mathfrak{g}}$ and $\widehat{\mathfrak{g}}$, since $d$ is not in the commutator subalgebra of $\widetilde{\mathfrak{g}}$. The action of $d$ defines a $\frac{1}{6}\mathbb{Z}$-grading on $V$. Denote the subspace of all homogeneous elements of degree $\frac{n}{6}$ by

$$V_n = \left\{ v \in V \,\middle|\, [d, v] = \frac{n}{6}\, v \right\}, \tag{3.2.2}$$

for $n \in \mathbb{Z}$, $n \leq 0$. It follows that $\deg v_0 = 0$ (since $\Lambda(d) = 0$), $\dim V_0 = 1$, $\dim V_n < \infty$ (for $n \leq 0$), and

$$V = \bigoplus_{n \leq 0} V_n.$$

**Definition 3.2.1.** The *graded dimension*, which we denote by $\chi_\Lambda(q)$, is a formal power series in the indeterminate $q$ with nonnegative integer coefficients:

$$\chi_\Lambda(q) = \sum_{n=0}^{\infty} (\dim V_{-n}) q^n \tag{3.2.3}$$

Let $\rho \in (\widetilde{\mathfrak{h}}_0)^*$ be such that $\rho(h_0) = \rho(h_1) = 1$ and $\phi = \Lambda + \rho$. Let

$$J_\Lambda = \left\{ n \in \mathbb{N} \,\middle|\, n \not\equiv \phi(c), \pm\phi(h_0), \pm\phi(h_1), \pm\phi(h_0 + h_1) \pmod{2\phi(c)} \right\}, \tag{3.2.4}$$

and

$$K_\Lambda = \begin{cases} \{\, n \in \mathbb{N} \mid \, n \equiv \phi(h_0) \pmod{2\phi(c)} \,\} & \text{if } \Lambda(h_0) = \Lambda(h_1), \\ \emptyset & \text{if } \Lambda(h_0) \neq \Lambda(h_1). \end{cases} \tag{3.2.5}$$

Using the numerator formula (see [LM78, Lep78]) and the principal specialization of the Weyl-Kac character formula, we have

$$\chi_\Lambda(q) = \prod_{\substack{n\in\mathbb{N} \\ n\equiv\pm1 \pmod 6}} (1-q^n)^{-1} \prod_{n\in J_\Lambda} (1-q^n)^{-1} \prod_{n\in K_\Lambda} (1-q^n). \tag{3.2.6}$$

Note that for level 4 standard modules of $\mathrm{A}_2^{(2)}$, we always have $\Lambda(h_0) \neq \Lambda(h_1)$.

**Proposition 3.2.2.** *The graded dimensions of the three level* 4 *standard modules for* $\mathrm{A}_2^{(2)}$ *are given by:*

**(4,0)-module:** $\Lambda = 4h_0^*$

$$\chi_{(4,0)}(q) = \chi_\Lambda(q) = \prod_{\substack{n\in\mathbb{N} \\ n\equiv\pm1 \ (\mathrm{mod}\ 6)}} (1-q^n)^{-1} \prod_{\substack{n\in\mathbb{N} \\ n\equiv\pm2,\pm3,\pm4 \\ (\mathrm{mod}\ 14)}} (1-q^n)^{-1}, \tag{3.2.7}$$

**(2,1)-module:** $\Lambda = 2h_0^* + h_1^*$

$$\chi_{(2,1)}(q) = \chi_\Lambda(q) = \prod_{\substack{n\in\mathbb{N} \\ n\equiv\pm1 \ (\mathrm{mod}\ 6)}} (1-q^n)^{-1} \prod_{\substack{n\in\mathbb{N} \\ n\equiv\pm1,\pm4,\pm6 \\ (\mathrm{mod}\ 14)}} (1-q^n)^{-1}, \tag{3.2.8}$$

**(0,2)-module:** $\Lambda = 2h_1^*$

$$\chi_{(0,2)}(q) = \chi_\Lambda(q) = \prod_{\substack{n\in\mathbb{N} \\ n\equiv\pm1 \ (\mathrm{mod}\ 6)}} (1-q^n)^{-1} \prod_{\substack{n\in\mathbb{N} \\ n\equiv\pm2,\pm5,\pm6 \\ (\mathrm{mod}\ 14)}} (1-q^n)^{-1}, \tag{3.2.9}$$

*Proof.* Follows from straightforward application of (3.2.6). □

Let $\Lambda = k_0h_0^* + k_1h_1^* \in (\widetilde{\mathfrak{h}}_0)^*$ be such that $k_0, k_1 \in \mathbb{Z}_{\geq 0}$ and $k_0 + 2k_1 = 4$. We will use the following notations for convenience.

**Notation 3.2.3.** (i) (For $\Lambda = 4h_0^*$)

$$I_{(4,0)} = I_\Lambda = \left\{ n \in \mathbb{N} \,\middle|\, n \equiv \pm2, \pm3, \pm4 \pmod{14} \right\}, \tag{3.2.10}$$

(ii) (For $\Lambda = 2h_0^* + h_1^*$)

$$I_{(2,1)} = I_\Lambda = \left\{ n \in \mathbb{N} \,\middle|\, n \equiv \pm 1, \pm 4, \pm 6 \pmod{14} \right\}, \tag{3.2.11}$$

(iii) (For $\Lambda = 2h_1^*$)

$$I_{(0,2)} = I_\Lambda = \left\{ n \in \mathbb{N} \,\middle|\, n \equiv \pm 2, \pm 5, \pm 6 \pmod{14} \right\}. \tag{3.2.12}$$

**Notation 3.2.4.** We define the following formal power series in $q$:

$$H(q) = \prod_{\substack{n \in \mathbb{N} \\ n \equiv \pm 1 \pmod{6}}} (1 - q^n)^{-1}, \tag{3.2.13}$$

and

$$F_\Lambda(q) = \prod_{n \in I_\Lambda} (1 - q^n)^{-1}, \tag{3.2.14}$$

for $\Lambda$ as above. We will also use the notation $F_\Lambda(n)$ (respectively, $H(n)$), for $n \geq 0$, to denote the coefficient of $q^n$ in $F_\Lambda(q)$ (respectively, $H(q)$).

*Remark.* $F_\Lambda(n)$ is the number of partitions of an integer $n \geq 0$ into parts from the set $I_\Lambda$.

Therefore, we can rewrite the graded dimension in Proposition 3.2.2 as

$$\chi_\Lambda(q) = H(q) F_\Lambda(q). \tag{3.2.15}$$

We will use the notation $\chi_\Lambda(n)$, $n \geq 0$, to denote the coefficient of $q^n$ in $\chi_\Lambda(q)$.

# Chapter 4

# The Framework

In this chapter, we lay out the framework—definitions, notations and a few results—on which the rest of this dissertation depends. We present a number of useful tools and techniques that we use repeatedly in the later discourse. The content of this chapter is valid for standard modules of any positive level. Throughout this chapter, let $V$ denote a standard module for $\widetilde{\mathfrak{g}}$ with highest weight $\Lambda$, and a highest weight vector $v_0$.

We would like to describe the elements of the universal enveloping algebra $\mathcal{U} = \mathcal{U}(\widetilde{\mathfrak{g}})$ as linear combinations of the "standard monomials" which are parametrized by certain partitions and "generalized partitions." We would also like to describe the structure of $V$ in terms of the action of the above mentioned monomials on the higest weight vector $v_0$.

In § 4.1, we present the definitions, notations, and a few results related to partitions and generalized partitions.

In § 4.2, we describe a standard monomial basis for the universal enveloping algebra $\mathcal{U}$, parametrized by the partitions and generalized partitions as defined in § 4.1. We also describe a filtration on $\mathcal{U}$ and present a few straightening lemmas in $\mathcal{U}$, which will be useful in the later discourse, with respect to this filtration.

In § 4.3, we describe two filtrations on $V$ and investigate the structure of $V$ in terms of the actions of the standard monomials (as described in § 4.2) on $v_0$ with respect to these filtrations. We give a spanning set for $V$, whose elements are parametrized by certain partitions, and show that a subset of this spanning set, parametrized by a certain restricted subset of partitions, is a basis for $V$. This enables us to "factor out" the factor $H(q)$ in the graded dimension formula (3.2.15) and only use the second factor $F_\Lambda(q)$ when comparing our spanning sets for "tightness" against the corresponding graded dimension.

In § 4.4, we describe a number of substantial tools and techniques that we use repeatedly in the later discourse. These tools and techniques are used to process various relations among the elements of the spanning sets (presented in § 4.3) and eliminate the extraneous elements thereof.

## 4.1 Tuples and Partitions

In this section, we set up the notations and definitions related to tuples, partitions and generalized partitions, and present a few related results which will be used later in this dissertation.

**Notation 4.1.1.** We denote the set of all tuples of integers by

$$\mathbb{Z}^* = \left\{ \mu = (m_1, \ldots, m_s) \,\middle|\, s \in \mathbb{Z}_{\geq 0},\ m_1, \ldots, m_s \in \mathbb{Z} \right\}.$$

Similarly, $\mathbb{N}^*$ denotes the set of all tuples of positive integers.

**Notation 4.1.2.** Let $\mu = (m_1, \ldots, m_s) \in \mathbb{Z}^s$ be any tuple of integers. We define the *length* and the *size* of $\mu$ by

$$l(\mu) = s, \tag{4.1.1}$$

$$|\mu| = \sum_{i=1}^{s} m_i, \tag{4.1.2}$$

respectively. For $1 \leq i \leq s$, we refer to $m_i$ as a *part* of $\mu$.

**Notation 4.1.3.** Let $\mu = (m_1, \ldots, m_s) \in Z^s$ and $\sigma \in \mathrm{Sym}(s)$ be a permutation of $\{1, \ldots, s\}$. Then define $\sigma(\mu)$ by

$$\sigma(\mu) = (m_{\sigma(1)}, \ldots, m_{\sigma(s)}).$$

**Notation 4.1.4.** For $\mu \in \mathbb{Z}^*$, we denote $\bar{\mu}$ as the rearrangement of $\mu$ in non-increasing order, i.e., $\bar{\mu} = (m'_1, \ldots, m'_s) = \sigma(\mu)$ for some permutation $\sigma$ of $\{1, \ldots, s\}$, such that $m'_1 \geq \cdots \geq m'_s$.

**Definition 4.1.5.** Define an equivalence relation $(\sim)$ on $\mathbb{Z}^*$ by

$$\mu_1 \sim \mu_2 \qquad \text{if } l(\mu_1) = l(\mu_2) \text{ and } \mu_2 = \sigma(\mu_1) \text{ for some } \sigma \in \mathrm{Sym}(l(\mu_1)),$$

for $\mu_1, \mu_2 \in \mathbb{Z}^*$.

**Definition 4.1.6** ((Generalized) Partition)**.** A *partition* of a nonnegative integer $n$ into positive parts is an equivalence class in $\mathbb{N}^*$ under $\sim$, such that $|\mu| = n$ for any $\mu$ in that equivalence class.

Similarly, a *generalized partition* of an integer $n$ is an equivalence class in $\mathbb{Z}^*$, such that $|\mu| = n$ for any $\mu$ in that equivalence class.

Thus, every (generalized) partition can be uniquely represented by a tuple arranged in non-increasing order. We will henceforth identify a partition (or a generalized partition) with a tuple of positive integers (or any integers) arranged in non-increasing order.

*Remark.* Notice that, for example, $(2), (2,0), (2,0,0)$, etc. are all considered different generalized partitions.

**Notation 4.1.7.** We denote the set of all partitions by

$$\mathscr{P} = \left\{ (m_1, \dots, m_s) \,\middle|\, s \in \mathbb{Z}_{\geq 0},\, m_i \in \mathbb{N} \text{ for all } 1 \leq i \leq s,\, m_1 \geq \dots \geq m_s \right\}. \tag{4.1.3}$$

Similarly, the set of all generalized partitions is denoted by

$$\mathscr{Q} = \left\{ (m_1, \dots, m_s) \,\middle|\, s \in \mathbb{Z}_{\geq 0},\, m_i \in \mathbb{Z} \text{ for all } 1 \leq i \leq s,\, m_1 \geq \dots \geq m_s \right\}. \tag{4.1.4}$$

We will also need the following subset of $\mathscr{P}$ later:

$$\mathscr{O} = \left\{ (m_1, \dots, m_s) \in \mathscr{P} \,\middle|\, m_1, \dots, m_s \equiv \pm 1 \pmod 6 \text{ for all } 1 \leq i \leq s \right\} \tag{4.1.5}$$

**Notation 4.1.8.** Let $\mathscr{X}$ be any subset of $\mathscr{Q}$ (e.g., $\mathscr{P}, \mathscr{Q}$, or $\mathscr{O}$). We will use the following notations (unless otherwise mentioned):

$$\mathscr{X}_s = \left\{ \mu \in \mathscr{X} \,\middle|\, l(\mu) = s \right\}, \tag{4.1.6}$$

$$\mathscr{X}(n) = \left\{ \mu \in \mathscr{X} \,\middle|\, |\mu| = n \right\}, \tag{4.1.7}$$

$$\mathscr{X}(n, s) = \mathscr{X}_s(n) = \mathscr{X}(n) \cap \mathscr{X}_s. \tag{4.1.8}$$

It will be sometimes useful to describe the partitions using "difference conditions."

**Definition 4.1.9** (Difference condition)**.** We say that a partition $\mu = (m_1, \ldots, m_s) \in \mathscr{P}$ satisfies the *difference condition*

$$\Delta = [d_1, \ldots, d_{s-1}],$$

if $m_i - m_{i+1} = d_i$, for $1 \leq i \leq s-1$.

We may add a "$\pm$" sign at the end of $d_1$ to denote if the first part is required to be even/odd. For example, $[3-, 3, 0]$ denotes the partitions satisfying the difference condition $[3, 3, 0]$ and having an odd entry as the first part. Therefore, the partitions $(9, 6, 3, 3)$, $(11, 8, 5, 5)$, $\ldots$, etc. satisfy $[3-, 3, 0]$, but the partition $(8, 5, 2, 2)$ does not.

Also, we may add a "$*$" as a superscript to an entry, say $d_i$, in the difference condition to denote zero or more occurrence of that entry. For example, $[3, 2^*, 3, 0]$ denotes the difference conditions where $2^*$ can be expanded to an arbitrary (including zero) number of 2's. Examples of partitions satisfying $[3, 2^*, 3, 0]$ include $(9, 6, 3, 3)$, $(11, 8, 6, 3, 3)$, $(13, 8, 6, 3, 3)$, $\ldots$, etc..

We now define a few operations on $\mathbb{Z}^*$.

**Definition 4.1.10** (Scaling)**.** For $\mu = (m_1, \ldots, m_s) \in \mathbb{Z}^s$ and $n \in \mathbb{Z}$ define the *scaling* of $\mu$ by $n$ as

$$n\mu = (n\mu_1, \ldots, n\mu_s). \tag{4.1.9}$$

**Definition 4.1.11** (Concatenation)**.** For $\mu_1 = (m_1, \ldots, m_r) \in \mathbb{Z}^r$ and $\mu_2 = (n_1, \ldots, n_s)$ $\in \mathbb{Z}^s$, define the *concatenation* of $\mu_1$ with $\mu_2$ as

$$\mu_1 \cdot \mu_2 = \mu_1\mu_2 = (m_1, \ldots, m_r, n_1, \ldots, n_s) \in \mathbb{Z}^{r+s}. \tag{4.1.10}$$

**Definition 4.1.12** (Translation)**.** For $(m_1, \ldots, m_s) \in \mathbb{Z}^s$ and $n \in \mathbb{Z}$, define the *translation* of $\mu$ by $n$ (we will typeset $n$ as boldface $\mathbf{n}$ for clarity) as

$$\mu + \mathbf{n} = (m_1 + n, \ldots, m_s + n). \tag{4.1.11}$$

**Definition 4.1.13** (Composition)**.** For $\mu_1, \mu_2 \in \mathbb{Z}^*$, define the *composition* of $\mu_1$ with $\mu_2$ as

$$\mu_1 \circ \mu_2 = \overline{\mu_1\mu_2} \in \mathscr{Q} \tag{4.1.12}$$

**Definition 4.1.14** (Sub-partition)**.** We say $\mu \in \mathbb{Z}^*$ is a *sub-tuple* of $\mu' \in \mathbb{Z}^*$, denoted $\mu \models \mu'$, if each part of $\mu$ appears in $\mu'$ with more or equal number of times than it appears in $\mu$.

We say $\mu$ is a *strict sub-tuple* of $\mu'$, denoted $\mu \vdash \mu'$, if $\mu \models \mu'$ and $\mu \neq \mu'$.

If both $\mu, \mu' \in \mathscr{Q}$ and $\mu \models \mu'$ (respectively, $\mu \vdash \mu'$), we say that $\mu$ is a *generalized sub-partition* (respectively, *strict generalized sub-partition*) of $\mu'$.

Similarly, if both $\mu, \mu' \in \mathscr{P}$ and $\mu \models \mu'$ (respectively, $\mu \vdash \mu'$), we say that $\mu$ is a *sub-partition* (respectively, *strict sub-partition*) of $\mu'$.

*Remark.* We will use the same symbols ("$\models$" or " $\vdash$") to denote both (strict) sub-tuple, generalized sub-partition or sub-partition—the only difference is in where $\mu, \mu'$ belong.

**Definition 4.1.15** (Prefix)**.** We say that $\mu \in \mathbb{Z}^r$ is a *prefix* of $\mu' \in \mathbb{Z}^s$, if $r \leq s$ and $\mu' = \mu\mu_1$ for some $\mu_1 \in \mathbb{Z}^{s-r}$. We say that $\mu$ is *strictly* a prefix of $\mu'$, if it is a prefix and $\mu \neq \mu'$.

**Definition 4.1.16** (Suffix)**.** Similarly, we say that $\mu \in \mathbb{Z}^r$ is a *suffix* of $\mu' \in \mathbb{Z}^s$, if $r \leq s$ and $\mu' = \mu_1\mu$ for some $\mu_1 \in \mathbb{Z}^{s-r}$. $\mu$ is *strictly* a suffix of $\mu'$, if it is a prefix and $\mu \neq \mu'$.

**Notation 4.1.17** (Lexicographic ordering)**.** For $\mu = (m_1, \ldots, m_r) \in \mathbb{Z}^r$ and $\mu' = (n_1, \ldots, n_s) \in \mathbb{Z}^s$, we say that $\mu$ is *lexicographically smaller* than $\mu'$ (denoted by $\mu \prec \mu'$) if either of the following holds:

(i) $\mu$ is strictly a prefix of $\mu'$, or

(ii) there is an $1 \leq i \leq \min(r, s)$ such that $m_1 = n_1, \ldots, m_{i-1} = n_{i-1}$ and $m_i < n_i$.

We use the following definition from [Cap88, Cap92, Cap93] to introduce a well-order on $\mathscr{P}$.

**Definition 4.1.18** (Ordering on $\mathbb{Z}^*$)**.** For $\mu, \mu' \in \mathbb{Z}^*$, we say that $\mu$ is *smaller* than $\mu'$ (denoted $\mu < \mu'$), if one of the following holds:

(i) $l(\mu) > l(\mu')$,

(ii) $l(\mu) = l(\mu')$ and $|\mu| > |\mu'|$,

(iii) $l(\mu) = l(\mu')$, $|\mu| = |\mu'|$ and $\mu \prec \mu'$.

*Remark.* The restriction of the relation $(<)$ on $\mathscr{P}$ is a (reverse) well-order, in the sense that every nonempty subset of $\mathscr{P}$ has a largest element. The empty partition $\varnothing$ is the largest element in $\mathscr{P}$. Therefore, one can use induction on $\mathscr{P}$. However, this is not a well-order on $\mathscr{Q}$.

**Notation 4.1.19.** Let $\mu_i^{(s)}(n)$ denote the $i$-th smallest partition in $\mathscr{P}_s(n)$ with respect to "$<$".

**Definition 4.1.20** (A partial order on $\mathbb{Z}^s$)**.** Let $\mu = (m_1, \ldots, m_s)$, $\mu' = (m'_1, \ldots, m'_s) \in \mathbb{Z}^s$. We will write $\mu \trianglelefteq \mu'$, if $m_i \leq m'_i$ for all $1 \leq i \leq s$. We will write $\mu \triangleleft \mu'$, if $\mu \trianglelefteq \mu'$ and $\mu \neq \mu'$.

*Remark.* Notice that, on $\mathbb{Z}^s$, $\mu \triangleleft \mu'$ implies that $\mu \prec \mu'$.

The following results about tuples and partitions will be used later to straighten out an out-of-order monomial in terms of the standard monomials defined in § 4.2.

**Lemma 4.1.21.** *For $\nu \in \mathbb{Z}^s$, $\nu \preceq \overline{\nu}$.*

*Proof.* Let $\nu = (n_1, \ldots, n_s)$ and $\overline{\nu} = (n'_1, \ldots, n'_s)$. If $\nu = \overline{\nu}$ then there is nothing to prove. Assume that $\nu \neq \overline{\nu}$. Let $k$ be the first index where they differ. Then we must have $n_k < n'_k$ (otherwise, it won't be out of place). Thus the result follows. ☐

**Lemma 4.1.22.** *Let $\mu \in \mathscr{Q}_s$ and $\nu \in \mathbb{Z}^s$. If $\mu \prec \nu$, then $\mu \prec \overline{\nu}$.*

*Proof.* We have $\mu \prec \nu \preceq \overline{\nu}$ (by Lemma 4.1.21). ☐

**Lemma 4.1.23.** *Let $\mu \in \mathscr{Q}_s$ and $\nu \in \mathbb{Z}^s$. If $\mu \triangleright \nu$, then $\mu \succ \overline{\nu}$.*

*Proof.* Let $\mu = (m_1, \ldots, m_s)$, $\nu = (n_1, \ldots, n_s)$ and $\overline{\nu} = (n'_1, \ldots, n'_s)$.

By the hypothesis, we have $m_1 \geq \cdots \geq m_s$, and $m_i \geq n_i$ for all $i \leq s$ with at least one strict inequality. Let $k$ be the first index where $\mu$ and $\nu$ differ. Therefore, $m_i = n_i$ for all $i < k$ and $m_k > n_k$. If $\nu = \overline{\nu}$, then we have the desired result.

Therefore, assume that $\nu \neq \overline{\nu}$. Then we have $n'_i = n_i$ for all $i < k$, since $n_i \geq n_j$ for all $i < k$ and $j \geq k$ ($n_i = m_i \geq m_k > n_k$, and $n_i = m_i \geq m_j \geq n_j$). This shows that $k < s$, since otherwise $\nu$ is already in non-increasing order (i.e., $\nu = \overline{\nu}$).

The above computation also shows that if $n_k$ is out of place, then it must be exchanged with an element of $\nu$ occurring further to the right, i.e., $n'_k = n_l > n_k$ for some $l > k$. However, if $n_k$ is not out of place, i.e., if $n'_k = n_k$, then we have the desired result, $\mu \succ \overline{\nu}$.

Therefore, we assume that $n_k$ is out of place, i.e., $n'_k = n_l > n_k$ for some $l > k$. Let $\nu'$ be the sequence obtained from $\nu$ by exchanging the $k$-th and the $l$-elements. Obviously, $\overline{\nu'} = \overline{\nu}$, and $\nu'$ also satisfies the original hypothesis that $\mu \rhd \nu'$, since $m_k \geq m_l \geq n_l$ and $m_l \geq n_l > n_k$.

If $m_k > n_l = n'_k$ then we have the result. If $m_k = n_l = n'_k$, then we repeat this argument using $\nu'$ in place of $\nu$. Note that the first place where $\mu$ and $\nu'$ differs is now after (greater than) $k$.

Thus, by finitely many application of the above argument, we arrive at the desired conclusion. $\square$

**Lemma 4.1.24.** *Let $\mu \in \mathscr{Q}$ and $\nu \in \mathbb{Z}^*$. If $\mu < \nu$, then $\mu < \overline{\nu}$.*

*Proof.* Notice that $l(\nu) = l(\overline{\nu})$ and $|\nu| = |\overline{\nu}|$. Therefore, if $\mu < \nu$ holds because of either $l(\mu) > l(\nu)$, or $l(\mu) = l(\nu)$ and $|\mu| > |\nu|$, then $\mu < \overline{\nu}$.

Therefore, assume that $l(\mu) = l(\nu)$, $|\mu| = |\nu|$ and $\mu \prec \nu$. Then the result follows from Lemma 4.1.22. $\square$

## 4.2 Standard Monomials

In this section, we present a standard monomial basis for the universal enveloping algebra $\mathcal{U} = \mathcal{U}(\widetilde{\mathfrak{g}})$, parametrized by certain partitions and generalized partitions. We also present a few straightening lemmas that will be useful later. First, we need to define the following elements in $\mathcal{U}$.

Fix any $\alpha \in L_2$. Let $X(n) = X(\alpha; n)$ (see Notation 2.2.5) and $E(n) = E_\alpha(n)$ (see Notation 2.2.2) for all $n \in \mathbb{Z}$.

**Notation 4.2.1.** For $\mu = (m_1, \ldots, m_s) \in \mathbb{Z}^s$, we define the elements $\alpha(\mu), E(\mu), X(\mu) \in$

$\mathcal{U}(\widetilde{\mathfrak{g}})$ by

$$\alpha(\mu) = \alpha(m_1)\ldots\alpha(m_s), \tag{4.2.1}$$

$$E(\mu) = E(m_1)\ldots E(m_s), \tag{4.2.2}$$

$$X(\mu) = X(m_1)\ldots X(m_s), \tag{4.2.3}$$

respectively.

*Remark.* Notice that $\alpha(\mu) = 0$, unless each part $m_i \equiv \pm 1 (\mathrm{mod}\ 6)$. Also note that each of these elements, unless zero, has degree $|\mu|/6$.

We have seen from Proposition 2.3.1 that the set

$$\left\{ \alpha(n) \,\middle|\, n \in \mathbb{Z},\, n \equiv \pm 1 \pmod 6 \right\} \bigcup \left\{ X(n) \,\middle|\, n \in \mathbb{Z} \right\} \bigcup \{c, d\} \tag{4.2.4}$$

spans $\widetilde{\mathfrak{g}}$. By PBW Theorem, the universal enveloping algebra $\mathcal{U} = \mathcal{U}(\widetilde{\mathfrak{g}})$ is the span of the monomials in these generators.

For convenience, we recall all the commutators (Lie brackets) of the generators in (4.2.4) in one place.

**Proposition 4.2.2.** *We have the following commutations in* $\widetilde{\mathfrak{g}}$*:*

(i) $c$ *is central.*

(ii) *For* $n \in \mathbb{Z}$, $[d, \alpha(n)] = \frac{n}{6}\alpha(n)$ *and* $[d, X(n)] = \frac{n}{6}X(n)$.

(iii) *For* $m, n \in \mathbb{Z}$ *with* $m, n \equiv \pm 1 (\mathrm{mod}\ 6)$, $[\alpha(m), \alpha(n)] = \frac{m}{6}\delta_{m+n,0}\, c$.

(iv) *For* $m, n \in \mathbb{Z}$ *with* $m \equiv \pm 1 (\mathrm{mod}\ 6)$, $[\alpha(m), X(n)] = X(m+n)$.

(v) *For* $m, n \in \mathbb{Z}$,

$$\begin{aligned} [X(m), X(n)] = {} & \frac{\omega^2}{6}\left(\omega^{n-m} - \omega^{m-n}\right) X(m+n) \\ & - \frac{\omega}{6}(-1)^m \alpha(m+n) \\ & + \delta_{m+n,0}\, \frac{\omega}{36}(-1)^m mc. \end{aligned} \tag{4.2.5}$$

*The coefficient to* $X(m+n)$ *simplifies to*

$$\frac{\omega^2}{6}\left(\omega^{n-m} - \omega^{m-n}\right) = \begin{cases} 0 & \text{if } n - m \equiv 0, 3 \pmod 6, \\ -\frac{\omega_0\sqrt{3}}{6} & \text{if } n - m \equiv 1, 2 \pmod 6, \\ \frac{\omega_0\sqrt{3}}{6} & \text{if } n - m \equiv 4, 5 \pmod 6). \end{cases}, \tag{4.2.6}$$

*where* $\omega = e^{\pi i/3}$ *and* $\omega_0 = e^{\pi i/6}$*.*

*Proof.* The first two brackets follow from the definition. For the last three see Proposition 2.2.13. $\square$

Using these commutators (Lie brackets) of the generators, any monomial (in the generators (4.2.4)) can be "straightened out" so as to express it as a linear combination of monomials of the form:

$$\alpha(-\lambda)X(-\mu)\alpha(\lambda')c^m d^n, \tag{4.2.7}$$

where $\lambda, \lambda' \in \mathscr{O}$, $\mu \in \mathscr{Q}$ and $m, n \in \mathbb{Z}_{\geq 0}$. These monomials are called the standard monomials. And the set of standard monomials form a basis for $\mathcal{U}(\mathfrak{g})$, called the standard monomial basis. Notice that $\alpha(-\lambda) \in \mathcal{U}(\mathfrak{s}_-)$ and $\alpha(\lambda') \in \mathcal{U}(\mathfrak{s}_+)$. We will refer to elements of $\mathfrak{s}_-\mathcal{U}(\mathfrak{s}_-)$ (respectively, $\mathfrak{s}_+\mathcal{U}(\mathfrak{s}_+)$) as *negative* (respectively, *positive*) *Heisenberg elements.*

We will use the following filtration on $\mathcal{U}$ (see [LW84]) to simplify the calculations in the straightening lemmas to be described later.

**Definition 4.2.3** ($\mathfrak{s}$-filtration on $\mathcal{U}$)**.** For $j \in \mathbb{Z}$, set

$$\mathcal{U}^{(j)} = 0 \quad \text{if } j < 0,$$
$$\mathcal{U}^{(0)} = \mathcal{U}(\mathfrak{s}),$$

and for $j > 0$,

$$\mathcal{U}^{(j)} = \text{Span}\left\{ x_1 \cdots x_n \in \mathcal{U} \;\middle|\; x_i \in \widetilde{\mathfrak{g}},\ \text{at most } j \text{ of the elements } x_r \text{ lie outside } \mathfrak{s} \right\},$$

where $\mathfrak{s}$ is the Heisenberg subalgebra as defined in (2.2.7a).

We clearly have

$$0 = \mathcal{U}^{(-1)} \subset \mathcal{U}^{(0)} \subset \mathcal{U}^{(1)} \subset \cdots \subset \mathcal{U} \tag{4.2.9}$$

and

$$\mathcal{U} = \bigcup_{j \geq 0} \mathcal{U}^{(j)}. \tag{4.2.10}$$

*Remark.* If we express an $x \in \mathcal{U}^{(s)}$, $s > 0$, as a linear combination of the standard monomials (4.2.7), then the number of $X(\bullet)$s appearing in each term can be at most $s$.

Now we present a few straightening lemmas in $\mathcal{U} = \mathcal{U}(\tilde{\mathfrak{g}})$.

**Lemma 4.2.4** (Straightening out $X(\bullet)$s)**.** *Let $\mu \in \mathbb{Z}^s$, and $\sigma \in \mathrm{Sym}(k)$ be a permutation of the indices $\{1, \ldots, s\}$. Then in $\mathcal{U}(\tilde{\mathfrak{g}})$, we have*

$$X(-\mu) - X(-\sigma\mu) \in \mathcal{U}^{(s-1)}. \tag{4.2.11}$$

*In particular,*

$$X(-\mu) \equiv X(-\bar{\mu}) \mod \mathcal{U}^{(s-1)}. \tag{4.2.12}$$

*Proof.* The result follows easily by multiple applications of Proposition 4.2.2(v). □

The following is a general fact which will be used on $\mathcal{U}$.

**Lemma 4.2.5.** *Let $\mathfrak{A}$ be any associative algebra. Let $y, x_1, \ldots, x_s \in \mathfrak{A}$. Then*

$$[y, x_1 \cdots x_s] = \sum_{i=1}^{s} x_1 \cdots x_{i-1}[y, x_i]x_{i+1} \cdots x_s. \tag{4.2.13}$$

*Proof.* The result follows from the fact that bracketing by $y$ is a derivation. □

The following lemmas shows how to straighten out an out-of-order monomial involving Heisenberg elements.

**Lemma 4.2.6** (Moving $\alpha(\pm n)$)**.** *Let $n \in N$ with $n \equiv \pm 1 (\mathrm{mod}\ 6)$ and $\mu \in \mathscr{Q}$. If $\mu = (m_1, \ldots, m_s)$, define $\mu(i^-) = (m_1, \ldots, m_{i-1})$ and $\mu(i^+) = (m_{i+1}, \ldots, m_s)$ (by convention, $\mu(1^-) = \mu(s^+) = \varnothing$). Then, we have*

$$X(-\mu)\alpha(-n) = \alpha(-n)X(-\mu) - \sum_{i=1}^{l(\mu)} X(-\mu(i^-))X(-(m_i + n))X(-\mu(i^+)), \tag{4.2.14}$$

*and*

$$\alpha(n)X(-\mu) = X(-\mu)\alpha(n) + \sum_{i=1}^{l(\mu)} X(-\mu(i^-))X(-(m_i - n))X(-\mu(i^+)). \tag{4.2.15}$$

*Proof.* The result follows from Lemma 4.2.5 and Proposition 4.2.2(iv). □

*Remark.* (i) When we move a single negative Heisenberg generator $\alpha(-n)$ to the left past $X(-\mu)$, we get a bunch of terms, for each of which exactly one part $m_i$, $1 \le i \le l(\mu)$, of $\mu$ gets increased by $n$. Thus, the resulting $\mu'$s that appear in the sum of the RHS of (4.2.14) (note that $\mu'$ could be out of order and not in $\mathscr{Q}$) satisfy the following properties: $\mu' \rhd \mu$, $\mu' \succ \mu$, $\overline{\mu'} \succ \mu$ (by Lemma 4.1.22).

(ii) When we move a single positive Heisenberg generator $\alpha(n)$ to the right past $X(-\mu)$, we get a bunch of terms, for each of which exactly one part $m_i$, $1 \le i \le l(\mu)$, of $\mu$ gets decreased by $n$. Thus, the resulting $\mu'$s that appear in the sum of the RHS of (4.2.15) (note that $\mu'$ could be out of order and not in $\mathscr{Q}$) satisfy the following properties: $\mu' \lhd \mu$, $\mu' \prec \mu$, $\overline{\mu'} \prec \mu$ (by Lemma 4.1.23).

**Lemma 4.2.7** (Moving $\alpha(-\lambda)$)**.** *Let $\lambda \in \mathscr{O}$ with $|\lambda| > 0$ and $\mu \in \mathscr{Q}_s(n)$. Then the monomial $X(-\mu)\alpha(-\lambda)$ can be straightened out in the form:*

$$X(-\mu)\alpha(-\lambda) = \sum_{\substack{\lambda' \in \mathscr{O},\, \lambda' \models \lambda \\ \mu' \in \mathbb{Z}^s\ \mu' \unrhd \mu \\ |\mu'| = |\mu| + |\lambda| - |\lambda'|}} a_{\lambda',\mu'}\, \alpha(-\lambda')X(-\mu') \tag{4.2.16}$$

$$\equiv \sum_{\substack{\mu'' \in \mathscr{Q}_s \\ \mu'' \succ \mu \\ |\mu''| = |\mu| + |\lambda|}} b_{\varnothing,\mu''} X(-\mu'') + \sum_{\substack{\lambda' \in \mathscr{O}, \\ \lambda' \models \lambda,\, \lambda' \neq \varnothing \\ \mu'' \in \mathscr{Q}_s,\, \mu'' \succeq \mu \\ |\mu''| = |\mu| + |\lambda| - |\lambda'|}} b_{\lambda',\mu''} \alpha(-\lambda')X(-\mu'') \mod \mathcal{U}^{(s-1)}, \tag{4.2.17}$$

*where $a_{\lambda',\mu'}$ and $b_{\lambda',\mu''}$ are constants with $a_{\lambda,\mu} = b_{\lambda,\mu} = 1$. Furthermore, the second equation* (4.2.17) *is obtained from the first* (4.2.16) *by rearranging out of order $X(-\mu')$s into $X(-\mu'')$s using Lemma 4.2.4, and we may take*

$$b_{\lambda',\mu''} = \sum a_{\lambda',\mu'}, \tag{4.2.18}$$

*where the sum is taken over all $\mu'$ appearing in the sum of* (4.2.16) *such that $\overline{\mu'} = \mu''$. The sums in* (4.2.16) *and* (4.2.17) *are finite.*

*Proof.* The first equation (4.2.16) follows easily from Lemma 4.2.6(4.2.14) by induction on $l(\lambda)$. Notice that the $\mu'$ that appear as a result of repeated application of (4.2.14), are gotten from $\mu$ by increasing various of its parts by various combinations of parts of $\lambda$.

We obtain the second equation (4.2.17) from the first (4.2.16) by straightening out each out-of-order $X(-\mu')$ occurring in the RHS of (4.2.16) into $X(-\mu'')$ (i.e., by taking $\mu'' = \overline{\mu'}$) using Lemma 4.2.4. Notice that by Lemma 4.1.22, all $\mu''$s that arise this way are lexicographically bigger than $\mu$, except when $\mu'' = \mu' = \mu$. Also notice that we have broken up the terms in (4.2.17) into two sums depending on whether they contain any negative Heisenberg element or not.

The rest of the assertion are obvious. □

**Lemma 4.2.8** (Moving $\alpha(\lambda)$)**.** *Let $\lambda \in \mathscr{O}$ with $|\lambda| > 0$ and $\mu \in \mathscr{Q}_s(n)$. Then the monomial $\alpha(\lambda)X(-\mu)$ can be straightened out in the form:*

$$\alpha(\lambda)X(-\mu) = \sum_{\substack{\lambda' \in \mathscr{O},\, \lambda' \models \lambda \\ \mu' \in \mathbb{Z}^s\, \mu' \trianglelefteq \mu \\ |\mu'| = |\mu| - |\lambda| + |\lambda'|}} a_{\lambda',\mu'}\, X(-\mu')\alpha(\lambda') \tag{4.2.19}$$

$$\equiv \sum_{\substack{\mu'' \in \mathscr{Q}_s \\ \mu'' \prec \mu \\ |\mu''| = |\mu| - |\lambda|}} b_{\varnothing,\mu''} X(-\mu'') + \sum_{\substack{\lambda' \in \mathscr{O}, \\ \lambda' \models \lambda,\, \lambda' \neq \varnothing \\ \mu'' \in \mathscr{Q}_s,\, \mu'' \preceq \mu \\ |\mu''| = |\mu| - |\lambda| + |\lambda'|}} b_{\lambda',\mu''} X(-\mu'')\alpha(\lambda') \mod \mathcal{U}^{(s-1)}, \tag{4.2.20}$$

*where $a_{\lambda',\mu'}$ and $b_{\lambda',\mu''}$ are constants with $a_{\lambda,\mu} = b_{\lambda,\mu} = 1$. Furthermore, the second equation* (4.2.20) *is obtained from the first* (4.2.19) *by rearranging out of order $X(-\mu')$s into $X(-\mu'')$s using Lemma 4.2.4, and we may take*

$$b_{\lambda',\mu''} = \sum a_{\lambda',\mu'}, \tag{4.2.21}$$

*where the sum is taken over all $\mu'$ appearing in the sum of* (4.2.19) *such that $\overline{\mu'} = \mu''$. The sums in* (4.2.19) *and* (4.2.20) *are finite.*

*Proof.* The first equation (4.2.19) follows easily from Lemma 4.2.6(4.2.15) by induction on $l(\lambda)$. Notice that the $\mu'$ that appear as a result of repeated application of (4.2.15), are gotten from $\mu$ by decreasing various of its parts by various combinations of parts of $\lambda$.

We obtain the second equation (4.2.20) from the first (4.2.19) by straightening out each out-of-order $X(-\mu')$ occurring in the RHS of (4.2.19) into $X(-\mu'')$ (i.e., by taking $\mu'' = \overline{\mu'}$) using Lemma 4.2.4. Notice that by Lemma 4.1.23, all $\mu''$s that arise this way are lexicographically smaller than $\mu$, except when $\mu'' = \mu' = \mu$. Also notice that we have broken up the terms in (4.2.20) into two sums depending on whether they contain any positive Heisenberg element or not.

The rest of the assertion are obvious. □

## 4.3 Structure of the Standard Modules

In this section, we will analyze the structure of $V$ in terms of the vertex operators (or the standard monomials described in § 4.2) acting on the highest weight vector $v_0$. We present a spanning set parametrized by the partitions in $\mathscr{O}$ and $\mathscr{P}$. We define a $\mathscr{P}$-filtration on $V$ based on the well-order "$>$" on $\mathscr{P}$, and an $\mathfrak{s}$-filtration on $V$ based on the $\mathfrak{s}$-filtration on $\mathcal{U}$. We also show the existence of a basis parametrized by partitions in $\mathscr{O}$ and irreducible partitions in $\mathscr{P}$. This enables us to use only the second factor $F_\Lambda(q)$ in the graded dimension formula (3.2.15) when comparing our spanning sets for "tightness." We also present a few results related to the action of various elements of $\mathcal{U}$ on $V$ with respect the filtrations defined on $V$.

Recall that $\mathfrak{n}_+.v_0 = 0$ and $\widetilde{\mathfrak{h}}_0 v_0 \in \mathbb{C}v_0$ (See § 3.1).

**Definition 4.3.1** ($\mathfrak{s}$-filtration on $V$)**.** For $s \in \mathbb{Z}_{\geq 0}$, define

$$V^{(s)} = \mathcal{U}^{(s)} v_0 \tag{4.3.1}$$

Clearly, $V^{(r)} \subset V^{(s)}$ if $r \leq s$, and $V = \bigcup_{s=0}^{\infty} V^{(s)}$.

**Lemma 4.3.2.** *Let $\lambda \in \mathscr{O}$ and $\mu \in \mathscr{Q}_s$.*

(i) *If $\lambda \neq \varnothing$ then $\alpha(\lambda)v_0 = 0$.*

(ii) *If $\mu$ contains a negative term then $X(-\mu)v_0 = 0$.*

(iii) *If $\mu$ contains 0 as a part then $X(-\mu)v_0 \in V^{(s-1)}$.*

*Proof.* (i) If $\lambda \in \mathscr{O}$ with $\lambda \neq \varnothing$, then we have $\alpha(\lambda) \in \mathfrak{n}_+\mathcal{U}(\mathfrak{n}_+)$.

(ii) Let $\mu = (m_1, \dots, m_s) \in \mathscr{Q}$ with at least one negative part. Since $\mu$ is arranged in non-increasing order, therefore $m_s < 0$. Therefore, $X(-m_s) \in \mathfrak{n}_+\mathcal{U}(\mathfrak{n}_+)$.

(iii) Let $\mu = (m_1, \dots, m_s) \in \mathscr{Q}$. Then $m_s \leq 0$ (since $\mu$ contains 0). If $m_s < 0$ then $X(-\mu)v_0 = 0$. If $m_s = 0$, then $X(-m_s) = X(0)$, and $X(0)v_0 \in \mathbb{C}v_0$. □

**Proposition 4.3.3.** *The set*

$$\left\{ \alpha(-\lambda)X(-\mu)v_0 \,\middle|\, \lambda \in \mathscr{O}, \mu \in \mathscr{P} \right\} \tag{4.3.2}$$

*is a spanning set for $V$.*

*Proof.* Since $v_0$ generates $V$ as a $\mathcal{U}(\widetilde{\mathfrak{g}})$-module, therefore $V = \mathcal{U}(\widetilde{\mathfrak{g}})v_0$. The result follows from Lemma 4.3.2 and the fact that the standard monomials (4.2.7) form a basis for $\mathcal{U}(\widetilde{\mathfrak{g}})$. Notice that $c$ acts as a scalar multiplication on $V$ and $dv_0 = 0$. □

**Lemma 4.3.4.** $V^{(s)} = \operatorname{Span}\{\, \alpha(-\lambda)X(-\mu)v_0 \mid \lambda \in \mathscr{O},\, \mu \in \mathscr{P},\, l(\mu) \leq s \,\}$.

*Proof.* The assertion follows immediately from the definition of $V^{(s)}$ (Definition 4.3.1), and Proposition 4.3.3. □

**Definition 4.3.5** (Filtration on $V$ by $\mathscr{P}$)**.** For $\mu \in \mathscr{P}$, set

$$V_{(\mu)} = \begin{cases} 0 & \text{if } \mu = \varnothing, \\ \displaystyle\sum_{\mu' > \mu} \mathcal{U}(\mathfrak{s}_-)X(-\mu')v_0 & \text{otherwise.} \end{cases} \tag{4.3.3}$$

Then we clearly have $V_{(\mu)} \subset V_{(\mu')}$ if $\mu \geq \mu'$, and

$$V = \bigcup_{\mu \in \mathscr{P}} V_{(\mu)}. \tag{4.3.4}$$

For brevity we will use the following terminologies.

**Definition 4.3.6** (Reducible)**.** A partition $\mu \in \mathscr{P}$, or the vector $X(-\mu)v_0$ is called *reducible* if $X(-\mu)v_0 \in V_{(\mu)}$. More generally, we say that $\mu$ is *reducible by partitions greater than* $\mu_0 \in \mathscr{P}$, if $X(-\mu)v_0 \in V_{(\mu_0)}$. We say that a partition is $\mu$ is *irreducible* if $\mu$ is not reducible, i.e., $X(-\mu)v_0 \notin V_{(\mu)}$.

**Notation 4.3.7.** Denote by

$$\mathscr{R}^{\Lambda} = \left\{ \mu \in \mathscr{P} \,\middle|\, X(-\mu)v_0 \in V_{(\mu)} \right\}, \tag{4.3.5}$$

the set of all reducible partitions, and

$$\mathscr{A}^{\Lambda} = \mathscr{P} \setminus \mathscr{R}^{\Lambda}, \tag{4.3.6}$$

the set of all irreducible partitions.

**Definition 4.3.8** (Forbidden)**.** A partition $\mu \in \mathscr{P}$ is called *forbidden*, if any partition having $\mu$ as a sub-partition is reducible, i.e., $X(-\overline{\mu\mu_*})v_0 \in V_{(\overline{\mu\mu_*})}$, for all $\mu_* \in \mathscr{P}$.

*Remark.* We will reduce the spanning set (4.3.2) further based on the structure of $V$. Later in this section we will show that for $\mu \in \mathscr{P}$, if $\mu$ is reducible then the elements $\alpha(-\lambda)X(-\mu)v_0$ can be removed from the spanning set (4.3.2) for all $\lambda \in \mathscr{O}$. We will also show that if we remove *all* such vectors from the spanning set, then the resulting subset is a basis.

**Lemma 4.3.9.** *If $\mu \in \mathscr{P}$, then $V^{(r)} \subset V_{(\mu)}$ for all $r < l(\mu)$.*

*Proof.* We have

$$\mathcal{U}^{(r)}v_0 = \sum_{\substack{\mu' \in \mathscr{P} \\ l(\mu') \leq r \\ \lambda \in \mathscr{O}}} \mathbb{C}\alpha(-\lambda)X(-\mu')v_0.$$

Each $\mu'$ in the above sum is larger than $\mu$ (since, $l(\mu') \leq r < l(\mu)$). Thus the result follows. □

The following proposition plays a very important role in the exposition later (cf. [Cap88, Cap92, MP87]).

**Proposition 4.3.10.** *Let $\mu \in \mathscr{Q}$ and $\mu' \in \mathbb{Z}^*$. If $\mu < \mu'$ then $X(-\mu')v_0 \in V_{(\mu)}$.*

*Proof.* Let $s = l(\mu') \leq l(\mu)$ (since $\mu' > \mu$). If $\mu' \in \mathscr{Q} \setminus \mathscr{P}$, then by Lemma 4.3.2 and Lemma 4.3.9,

$$X(-\mu')v_0 \in V^{(s-1)} \subset V_{(\mu)}.$$

Assume that $\mu' \in \mathscr{P}$. By Lemma 4.1.24, $\mu < \overline{\mu'}$. Therefore,

$$X(-\overline{\mu'})v_0 \in V_{(\mu)}.$$

By Lemma 4.2.4,

$$X(-\mu')v_0 \equiv X(-\overline{\mu'})v_0 \mod V^{(s-1)}.$$

Using Lemma 4.3.9 and the fact that $s \leq l(\mu)$, we have $V^{(s-1)} \subset V_{(\mu)}$. Therefore, we have $X(-\mu')v_0 \in V_{(\mu)}$. □

We will now proceed to show that all elements of the form $\alpha(-\lambda)X(-\mu)v_0$, where $\lambda \in \mathscr{O}$ and $\mu \in \mathscr{P}$ is reducible (i.e., $\mu \in \mathscr{R}^{\Lambda}$) can be removed from the spanning set (4.3.2). In fact, we will show that the resulting set (as described below) is a basis.

**Notation 4.3.11.** Recall Notation 4.3.7. We define

$$B_\Lambda = \left\{ \alpha(-\lambda)X(-\mu)v_0 \,\middle|\, \lambda \in \mathscr{O},\, \mu \in \mathscr{A}_\Lambda \right\}. \tag{4.3.7}$$

But, first, we need a few auxiliary lemmas.

**Proposition 4.3.12.** *The action of the Heisenberg subalgebra preserves the $\mathscr{P}$-filtration on $V$, i.e.,*

$$\mathcal{U}(\mathfrak{s})V_{(\mu)} \subset V_{(\mu)}, \tag{4.3.8}$$

*for any $\mu \in \mathscr{P}$. Furthermore, for any $\mu \in \mathscr{P}$,*

$$\mathfrak{s}_+ X(-\mu)v_0 \in V_{(\mu)}. \tag{4.3.9}$$

*Proof.* We first show the second assertion (4.3.9). Take any $n > 0$. It is enough to show that $\alpha(n)X(-\mu)v_0 \in V_{(\mu)}$. Using Lemma 4.2.6 (4.2.15), $\alpha(n)X(-\mu)v_0$ can be written as a finite sum of vectors of the form $X(-\mu')v_0$, where $\mu' \lhd \mu$. If $\mu'$ contains a non-positive entry then, by Lemma 4.3.2, $X(-\mu')v_0 \in V^{(s-1)}$, where $s = l(\mu)$. Otherwise, $\mu' \in \mathscr{P}$, $l(\mu') = l(\mu)$, $|\mu'| < |\mu|$ and therefore, $\mu' > \mu$. In both cases, $X(-\mu')v_0 \in V_{(\mu)}$ (using Lemma 4.3.9 in the first case, and Lemma 4.1.24 in the second). This proves the result.

Now we will prove the first assertion (4.3.8). Clearly, $\mathcal{U}(\mathfrak{s}_-)V_{(\mu)} \subset V_{(\mu)}$, by the definition of $V_{(\mu)}$ (Definition 4.3.5). Take any $n > 0$, $\lambda \in \mathscr{O}$ and $\mu' \in \mathscr{P}$ with $\mu' > \mu$. It is enough to show that $\alpha(n)\alpha(-\lambda)X(-\mu')v_0 \in V_{(\mu)}$.

Let $m$ be the number of times $n$ appears as a part in the partition $\lambda$. Then, using the bracket formula Proposition 4.2.2 (iii), we have

$$[\alpha(n), \alpha(-\lambda)] = \frac{mn}{6} c\alpha(-\lambda'), \tag{4.3.10}$$

where $\lambda'$ is the partition obtained from $\lambda$ by deleting one occurrence of the part $n$, and $c$ is the central element in $\mathfrak{s}$ acting on $V$ as the scalar $\Lambda(c)$

Therefore,

$$\alpha(n)\alpha(-\lambda)X(-\mu')v_0 = \alpha(-\lambda)\alpha(n)X(-\mu')v_0 + \frac{mn}{6}c\alpha(-\lambda')X(-\mu')v_0. \tag{4.3.11}$$

The second term $\alpha(-\lambda')X(-\mu') \in V_{(\mu)}$, since $\mu' > \mu$. For the first term, notice that $\alpha(n)X(-\mu')v_0 \in V_{(\mu')} \subset V_{(\mu)}$ by (4.3.9), and therefore, $\alpha(-\lambda)\alpha(n)X(-\mu')v_0 \in V_{(\mu)}$. This completes the proof. $\square$

We will need the following elementary lemma.

**Lemma 4.3.13.** *Let $\lambda_1 \neq \lambda_2$ be two distinct partitions of a positive integer $n$. Then there exists a part which occurs in $\lambda_1$ more often than it occurs in $\lambda_2$.*

*Proof.* Let $m_i(j)$ denote the number of times $j$ occurs as a part in the partition $\lambda_i$, $i = 1, 2$; $1 \leq j \leq n$. (Take $m_i(j) = 0$ if $j$ does not appear as a part in $\lambda_i$). Then, we have

$$n = \sum_{j=1}^{n} m_1(j)j = \sum_{j=1}^{n} m_2(j)j. \tag{4.3.12}$$

Assume, to the contrary, that $m_1(j) \leq m_2(j)$, for all $1 \leq j \leq n$. By (4.3.12), we must have $m_1(j) = m_2(j)$ for all $j$. This contradicts our assumption that $\lambda_1 \neq \lambda_2$. $\square$

**Lemma 4.3.14.** *Let $\lambda \in \mathscr{O}$.*

(1) $\alpha(\lambda)\alpha(-\lambda)X(-\mu)v_0 \equiv CX(-\mu)v_0 \mod V_{(\mu)}$, *for some constant $C \neq 0$.*

(2) *If $\lambda_0 \neq \lambda$ with $|\lambda_0| = |\lambda|$, then $\alpha(\lambda_0)\alpha(-\lambda)X(-\mu)v_0 \in V_{(\mu)}$,*

*Proof.* (1) We prove this by induction on $l(\lambda)$. If $\lambda = \varnothing$, then the statement is vacuously true with $C = 1$.

Assume that $l(\lambda) > 0$. Let $n_1, n_2, \ldots, n_s$ be the parts appearing in $\lambda$ with multiplicities $k_1, \ldots, k_s > 0$. If $s > 1$, then we have

$$\begin{aligned} \alpha(\lambda)\alpha(-\lambda)X(-\mu)v_0 &= \alpha(n_1)^{k_1}\alpha(-n_1)^{k_1} \cdots \alpha(n_s)^{k_s}\alpha(-n_s)^{k_s}X(-\mu)v_0 \\ &\equiv CX(-\mu)v_0 \mod V_{(\mu)} \end{aligned} \tag{4.3.13}$$

for some $C \neq 0$, by repeated application of the induction hypothesis.

Therefore, we may assume that $\lambda = (n, \ldots, n)$, with $n$ appearing $k$ times. Let $\lambda'$ be the partition with only part $n$ appearing $k - 1$ times. We have

$$\begin{aligned} \alpha(\lambda)\alpha(-\lambda)X(-\mu)v_0 &= \alpha(\lambda')\alpha(n)\alpha(-\lambda)X(-\mu)v_0 \\ &= \alpha(\lambda')\alpha(-\lambda)\alpha(n)X(-\mu)v_0 + \frac{kn}{6}c\alpha(\lambda')\alpha(-\lambda')X(-\mu)v_0 \\ &\equiv 0 + CX(-\mu)v_0 \mod V_{(\mu)}, \end{aligned} \tag{4.3.14}$$

for some $C \neq 0$, using (4.3.11), (4.3.9) and the induction hypothesis.

(2) Let $n$ be a part in $\lambda_0$ which appears $k_0$ times in $\lambda_0$, and $k$ times in $\lambda$ ($k$ may be 0), such that $k_0 > k$. Such a part exists by Lemma 4.3.13. Let $\lambda_0'$ be the partition obtained from $\lambda_0$ by deleting all occurrences of $n$. Similarly, let $\lambda'$ be the partition obtained from $\lambda$ by deleting all occurrences of $n$, if any. Now,

$$\alpha(\lambda_0)\alpha(-\lambda)X(-\mu)v_0 = \alpha(\lambda_0')\alpha(-\lambda')\alpha(n)^{k_0-k}\alpha(n)^k\alpha(-n)^kX(-\mu)v_0 \tag{4.3.15}$$

Using part (1), $\alpha(n)^k\alpha(-n)^kX(-\mu)v_0 \equiv CX(-\mu)v_0 \mod V_{(\mu)}$ for some $C \neq 0$. Using (4.3.9),

$$\begin{aligned} \alpha(n)^{k_0-k}\alpha(n)^k\alpha(-n)^kX(-\mu)v_0 &\equiv C\alpha(n)^{k_0-k}X(-\mu)v_0 \\ &\equiv 0 \mod V_{(\mu)}. \end{aligned} \tag{4.3.16}$$

Applying Proposition 4.3.12 (4.3.8), we obtain

$$\alpha(\lambda_0)\alpha(-\lambda)X(-\mu)v_0 \in V_{(\mu)}. \tag{4.3.17}$$

$\square$

Recall Notation 4.3.7 and Notation 4.3.11.

**Proposition 4.3.15.** *The set*

$$B_\Lambda = \left\{ \alpha(-\lambda)X(-\mu)v_0 \,\middle|\, \lambda \in \mathscr{O}, \mu \in \mathscr{A}^\Lambda \right\}.$$

*is a basis for* $V = L(\Lambda)$

*Proof of Proposition 4.3.15.* Let $S$ be the spanning set of $V$ as given in (4.3.2),

$$S = \left\{ \alpha(-\lambda)X(-\mu)v_0 \,\middle|\, \lambda \in \mathscr{O}, \mu \in \mathscr{P} \right\}. \tag{4.3.18}$$

First, we show the linear independence of $B_\Lambda$. If $B_\Lambda$ is not linearly independent then there exists a relation of vectors in $B_\Lambda$. Because $V$ is graded, we may assume that all vectors appearing in this relation are homogeneous of degree $-n/6$, for some $n > 0$. We write this relation as

$$\sum_{\substack{\lambda\in\mathscr{O},\mu\in\mathscr{A}^\Lambda \\ |\lambda|+|\mu|=n}} c_{\lambda,\mu}\alpha(-\lambda)X(-\mu)v_0 = 0, \tag{4.3.19}$$

where $c_{\lambda,\mu}$ are constants. Let $\mu_0$ be the least partition in $\mathscr{A}^\Lambda$ that appears in the above relation with nonzero coefficient. Clearly, $\mu_0 \neq \varnothing$, since the set $\{\, \alpha(-\lambda)v_0 \mid \lambda \in \mathscr{O} \,\}$ is linearly independent. Let $n_0 = |\mu_0| > 0$. Let

$$\mathscr{O}^{\mu_0} = \left\{ \lambda \in \mathscr{O} \,\middle|\, c_{\lambda,\mu_0} \neq 0 \right\} \subset \mathscr{O}(n - n_0). \tag{4.3.20}$$

Then (4.3.19) can be expressed as

$$\sum_{\lambda \in \mathscr{O}^{\mu_0}} c_\lambda \alpha(-\lambda) X(-\mu_0) v_0 \equiv 0 \mod V_{(\mu_0)}, \tag{4.3.21}$$

where $c_\lambda = c_{\lambda,\mu_0} \neq 0$. If $\mathscr{O}^{\mu_0} = \{\varnothing\}$, then (4.3.21) reduces to $c_\varnothing X(-\mu_0)v_0 \equiv 0 \mod V_{(\mu_0)}$. Therefore $X(-\mu_0)v_0 \in V_{(\mu_0)}$, contradicting our assumption that $\mu_0 \in \mathscr{A}^\Lambda$. Therefore, assume that $n - n_0 > 0$.

Choose $\lambda_0 \in \mathscr{O}^{\mu_0}$ arbitrarily. We multiply $\alpha(\lambda_0)$ to the left of (4.3.21). Using Lemma 4.3.14, we have

$$\sum_{\lambda \in \mathscr{O}^{\mu_0}} c_\lambda \alpha(\lambda_0) \alpha(-\lambda) X(-\mu_0) v_0 \equiv c_{\lambda_0} C X(-\mu_0) v_0 \mod V_{(\mu_0)}, \tag{4.3.22}$$

for some $C \neq 0$. Thus we arrive at a contradiction that $X(-\mu_0)v_0 \in V_{(\mu_0)}$ (or, equivalently, $\mu_0 \in \mathscr{R}^\Lambda$). This completes the proof of linear independence of $B_\Lambda$.

Now it remains to prove that $B_\Lambda$ is a spanning set. This is obvious since, we obtain $B_\Lambda$ by removing elements of $S$ using linear relations. However, we can give an alternative proof using induction on the well-ordered set $(\mathscr{P}, >)$.

It is enough to show that every vector $v = \alpha(-\lambda)X(-\mu)v_0 \in S$ can be expressed as a linear combination of vectors in $B_\Lambda$, for any $\lambda \in \mathscr{O}$ and $\mu \in \mathscr{P}$. The base case for our induction, $\mu = \varnothing$, is trivial, since $\alpha(-\lambda)v_0 \in B$.

Fix a $\mu \in \mathscr{P}$ and assume the result for $\mu' > \mu$. If $\mu \in \mathscr{A}^\Lambda$ there is nothing to prove. Assume that $\mu \in \mathscr{R}^\Lambda$. But then $v \in V_{(\mu)}$. Therefore, by induction hypothesis, $v$ is in the span of $B_\Lambda$. □

Assume that $V = L(\Lambda)$ is a level 4 standard module with highest weight $\Lambda$, and a highest weight vector $v_0$. We finish this section with a few useful observations related to the graded dimension (3.2.15) of $V$,

$$\chi_\Lambda(q) = H(q) F_\Lambda(q). \tag{4.3.23}$$

Let $\chi_\Lambda(n)$, $H(n)$ and $F_\Lambda(n)$ be the coefficient of $q^n$ in $\chi_\Lambda(q)$, $H(q)$ and $F_\Lambda(q)$ respectively, for $n \geq 0$. (see Notation 3.2.4). Let $B_\Lambda(q)$ be the generating function counting the number of elements in $B_\Lambda$ of degree $-n/6$,

$$B_\Lambda(q) = \sum_{n\geq 0} B_\Lambda(n)q^n. \tag{4.3.24}$$

Also recall Notation 4.1.8 and Notation 4.3.7.

**Lemma 4.3.16.** *Let $n$ be a positive integer. Then, we have*

(1) $\chi_\Lambda(n) = B_\Lambda(n)$,

(2) $F_\Lambda(n) = |\mathscr{A}^\Lambda(n)|$.

*Proof.* The first equality is obvious, since $B_\Lambda$ is a basis for $V$ by Proposition 4.3.15.

For the second equality, observe that

$$B_\Lambda(q) = \left(\sum_{n\geq 0} \mathscr{O}(n)q^n\right)\left(\sum_{n\geq 0} |\mathscr{A}^\Lambda(n)|q^n\right) = H(q)\sum_{n\geq 0} |\mathscr{A}^\Lambda(n)|q^n \tag{4.3.25}$$

Since $B(q) = X_\Lambda(q) = H(q)F_\Lambda(q)$, the result follows by canceling out the common factor $H(q)$. □

**Corollary 4.3.17.** *Let $S'_\Lambda$ be a subset of the spanning set* (4.3.2) *of $V = L(\Lambda)$, given by*

$$S'_\Lambda = \left\{ \alpha(-\lambda)X(-\mu)v_0 \,\middle|\, \lambda \in \mathscr{O}, \mu \in \mathscr{P}'_\Lambda \right\}, \tag{4.3.26}$$

*where $\mathscr{A}^\Lambda \subset \mathscr{P}'_\Lambda \subset \mathscr{P}$. Then, for every $n \geq 0$, we have*

$$F_\Lambda(n) \leq \left|\mathscr{P}'_\Lambda(n)\right|. \tag{4.3.27}$$

*Furthermore, if the equality holds for every $n \geq 0$, then $\mathscr{P}'_\Lambda = \mathscr{A}^\Lambda$ and $S'_\Lambda$ is a basis for $V$.*

*Proof.* This is an obvious consequence of Lemma 4.3.16 and the fact that $\mathscr{A}^\Lambda \subset \mathscr{P}'_\Lambda$. □

## 4.4 Tools and Techniques for Working with Relations

In this section, we describe a few tools and techniques that we will be using to discover reducible partitions from various relations among vectors in the spanning set (4.3.2).

These relations can be classified into two categories. The relations coming from various generating function identities, presented in Chapter 5, are valid for all level 4 standard modules. For each generating function, the coefficient of $z^{n/6}$, $n \geq 0$, gives us a family of relations among homogeneous operators on $V$ of degree $-n/6$. Applying these relations on $v_0$ (or, on $X(-\mu_*)v_0$), we obtain a family of relations for the vectors in the spanning set. The reducible partitions that arise this way are, in fact, "forbidden" partitions, in the sense that they cannot occur anywhere as a sub-partition in an irreducible partition. There are, however, relations among the spanning set vectors that are not coming from the operator identities. These relations are specific to a particular standard module of level 4. We describe these relations in Chapter 7.

We will first show that any partition ending with a reducible partition is also reducible. Then, we will investigate conditions that we need on the operator identities that give rise to "forbidden" partitions.

Even though we will be applying the tools and techniques for the level 4 standard modules, the arguments presented here are valid for any standard module. The computations shown here are to be thought of taking place in $\bar{\mathcal{U}}$, the image of the universal enveloping algebra $\mathcal{U}$ in $\operatorname{End} V$ via the representation $\mathcal{U} \to \operatorname{End} V$.

The following result shows that if $\mu_0 \in \mathscr{P}$ is reducible, then any partition $\mu \in \mathscr{P}$ ending with $\mu_0$ is also reducible.

**Proposition 4.4.1.** *Let $\mu_0, \mu_1 \in \mathscr{P}$ such that $\mu = \mu_1\mu_0 \in \mathscr{P}$. If $\mu_0$ is reducible then so is $\mu$, i.e.,*

$$X(-\mu_0)v_0 \in V_{(\mu_0)} \implies X(-\mu)v_0 \in V_{(\mu)}. \tag{4.4.1}$$

*Proof.* Assume that $X(-\mu_0)v_0 \in V_{(\mu_0)}$. Then, we may write

$$X(-\mu_0)v_0 = \sum_{\substack{\lambda \in \mathscr{O} \\ \mu' \in \mathscr{P}, \mu' > \mu_0 \\ |\mu'| = |\mu_0| - |\lambda|}} a_{\lambda,\mu'}\, \alpha(-\lambda)X(-\mu')v_0,$$

where $a_{\lambda,\mu'}$ are constants. We apply $X(-\mu_1)$ on both sides of the above equation.

Notice that, if $l(\mu') < l(\mu_0)$, then clearly the term

$$X(-\mu_1)\alpha(-\lambda)X(-\mu')v_0 \in V^{(l(\mu)-1)} \subset V_{(\mu)}.$$

Therefore, assume that $l(\mu') = l(\mu_0)$. We need to straighten out $X(-\mu_1)\alpha(-\lambda)$ using Lemma 4.2.7. Therefore, the term $X(-\mu_1)\alpha(-\lambda)X(-\mu')v_0$ can be expressed as

$$X(-\mu_1)\alpha(-\lambda)X(-\mu')v_0 = \sum_{\substack{\lambda'\in\mathscr{O},\lambda'\models\lambda \\ \mu''\in\mathscr{Q}_s,\,\mu''\succeq\mu_1 \\ |\mu''|=|\mu_1|+|\lambda|-|\lambda'|}} b_{\lambda',\mu''}\,\alpha(-\lambda')X(-\mu'')X(-\mu')v_0,$$

where $s = l(\mu_1)$. Notice that $l(\mu''\mu') = l(\mu_1\mu_0) = l(\mu)$. If $\lambda' \neq \varnothing$, then we have

$$|\mu''| + |\mu'| < |\mu_1| + |\lambda| + |\mu'| = |\mu_1| + |\mu_0| = |\mu|,$$

and hence, $\mu''\mu' > \mu$. If $\lambda' = \varnothing$, then we have $|\mu''\mu'| = |\mu|$ and $\mu''\mu' \succ \mu_1\mu_0 = \mu$. Thus, in either case $\mu''\mu' > \mu$. By Proposition 4.3.10, the terms

$$\alpha(-\lambda')X(-\mu'')X(-\mu')v_0 \in V_{(\mu)}.$$

This completes the proof. $\square$

*Remark.* Notice that in the hypothesis of the above Proposition 4.4.1 we require that $\mu = \mu_1\mu_0 \in \mathscr{P}$, i.e., $\mu$ is in non-increasing order. This condition can not be relaxed by replacing $\mu = \overline{\mu_1\mu_0}$. For example, if $X(-6)v_0 = \alpha(-1)X(-5)v_0$, then after applying $X(-1)$ and rearranging we get

$$X(-6,-1)v_0 \equiv \alpha(-1)X(-5,-1)v_0 - X(-5,-2)v_0 \mod V^{(1)}.$$

However $(5,2) \not> (6,1)$.

Now, we proceed to investigate the criteria on an operator identity so that the relation gives rise to "forbidden" partitions. We start with a few straightening lemmas.

**Notation 4.4.2.** Throughout this section, let $T(-n) \in \bar{\mathscr{U}}^{(s)}$ denote an arbitrary standard monomial of degree ${-n}/{6}$, i.e.,

$$T(-n) = \alpha(-\lambda_1)X(-\mu)\alpha(\lambda_2) \in \bar{\mathscr{U}}^{(s)} \tag{4.4.2}$$

for some $\lambda_1, \lambda_2 \in \mathscr{O}$ and $\mu \in \mathscr{P}$ such that

$$l(\mu) \leq s, \tag{4.4.3}$$

$$|\mu| + |\lambda_1| - |\lambda_2| = n. \tag{4.4.4}$$

**Notation 4.4.3.** We fix $\mu_0 \in \mathscr{P}_s(n)$ and $\mu_* \in \mathscr{P}$. Let

$$\widetilde{\mu} = \overline{\mu_0 \mu_*}, \tag{4.4.5}$$

$$t = l(\widetilde{\mu}). \tag{4.4.6}$$

We will now describe the action of the standard monomial $T(-n)$ on a vector $v = X(-\mu_*)v_0$ modulo the subspace $V_{(\widetilde{\mu})}$. But, first, we need the following elementary observation.

**Lemma 4.4.4.** *Let $\mu_1, \mu_2, \mu_* \in \mathscr{Q}$ with $\mu_1 \prec \mu_2$. $\overline{\mu_1 \mu_*} \prec \overline{\mu_2 \mu_*}$.*

*Proof.* We can characterize the lexicographic ordering in terms of the multiplicity of each part as follows.

Let $m_\mu(k)$ denote multiplicity of $k$ as a part in $\mu$. Note that $m_\mu(k)$ is taken to be 0 if $k$ does not appear in $\mu$ as a part.

Then $\nu_1 \prec \nu_2$ holds if and only if $m_{\nu_1}(k_0) < m_{\nu_2}(k_0)$, where

$$k_0 = \max \left\{ k \in \mathbb{Z} \,\middle|\, m_{\nu_1}(k) \neq m_{\nu_2}(k) \right\}.$$

This is obvious, since $\nu_1, \nu_2$ are arranged in non-increasing order.

Now, notice that for any $\mu \in \mathscr{Q}$,

$$m_{\overline{\mu\mu_*}}(k) = m_\mu(k) + m_{\mu_*}(k),$$

for all $k$. Thus, the conclusion is obvious. □

**Lemma 4.4.5.** *Recall Notation 4.4.3. Let $T(-n)$ be as defined in Notation 4.4.2 such that $\mu$ contains a non-positive integer as a part. Then*

$$T(-n)X(-\mu_*)v_0 \in V^{(t-1)} \subset V_{(\widetilde{\mu})}. \tag{4.4.7}$$

*Proof.* This follows immediately from Lemma 4.3.2. □

**Lemma 4.4.6.** *Recall Notation 4.4.3. Let $\mu \in \mathscr{P}$ such that $\mu > \mu_0$. Then*

$$X(-\mu)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})} \tag{4.4.8}$$

*Proof.* The conclusion is obvious if either $l(\mu) < l(\mu_0)$, or $l(\mu) = l(\mu_0)$ and $|\mu| < |\mu_0|$. Therefore, assume that $l(\mu) = l(\mu_0)$, $|\mu| = |\mu_0|$ and $\mu \succ \mu_0$.

By Lemma 4.4.4, we have $\overline{\mu\mu_*} \succ \overline{\mu_0\mu_*} = \widetilde{\mu}$. Thefore, the result follows after rearranging the $X(\bullet)$ operators using Lemma 4.2.4. $\square$

**Lemma 4.4.7.** *Recall Notation 4.4.3. Let $T(-n)$ be as defined in Notation 4.4.2, such that $\lambda_1 \neq \varnothing$. Then*

$$T(-n)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})}. \tag{4.4.9}$$

*Proof.* If $l(\mu) < l(\mu_0)$, then the result is obvious. Assume that $l(\mu) = l(\mu_0) = s$. We have,

$$T(-n) = \alpha(-\lambda_1)X(-\mu)\alpha(\lambda_2), \tag{4.4.10}$$

$$|\mu| = n - |\lambda_1| + |\lambda_2| < n + |\lambda_2|, \tag{4.4.11}$$

since $|\lambda_1| > 0$ by assumption. Note that $|\mu_0| = n$ by assumption in Notation 4.4.3.

The vector $T(-n)X(-\mu_*)v_0$ can be straightened out, using Lemma 4.2.8, in the form

$$T(-n)X(-\mu_*)v_0 = \sum_{\substack{\mu' \in \mathscr{Q}, \mu' \prec \mu_* \\ l(\mu') = l(\mu_*) \\ |\mu'| = |\mu_*| - |\lambda_2|}} b_{\varnothing,\mu'}\, \alpha(-\lambda_1)X(-\mu)X(-\mu')v_0. \tag{4.4.12}$$

Notice that $l(\mu\mu') = l(\mu_0\mu*) = l(\widetilde{\mu})$ and

$$|\mu\mu'| = |\mu| + |\mu'| < n + |\lambda_2| + |\mu'| = |\mu_0\mu_*|. \tag{4.4.13}$$

Therefore $\mu\mu_* > \widetilde{\mu}$. The conclusion follows from Proposition 4.3.10 $\square$

**Lemma 4.4.8.** *Let $T(-n) = X(-\mu)\alpha(\lambda_2)$ be as defined in Notation 4.4.2 with $\lambda_1 = \varnothing$, $\lambda_2 \neq \varnothing$, and either $l(\mu) < l(\mu_0) = s$ or $\mu \succ \mu_0$. Recall Notation 4.4.3. Then*

$$T(-n)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})}. \tag{4.4.14}$$

*Proof.* If $l(\mu) < l(\mu_0)$, then the result is obvious.

Assume that $l(\mu) = l(\mu_0) = s$. Using Lemma 4.2.8, We can express $T(-n)X(-\mu_*)v_0$ as

$$T(-n)X(-\mu_*)v_0 = \sum_{\substack{\mu' \in \mathscr{Q}, \mu' \prec \mu_* \\ l(\mu') = l(\mu_*) \\ |\mu'| = |\mu_*| - |\lambda_2|}} b_{\varnothing,\mu'} X(-\mu)X(-\mu')v_0 \tag{4.4.15}$$

Since, $\mu \succ \mu_0$, we have $\mu\mu' \succ \mu_0\mu_*$. By Lemma 4.4.4, $\overline{\mu\mu'} \succ \widetilde{\mu}$. Since, $l(\mu\mu') = l(\widetilde{\mu})$ and $|\mu\mu'| = |\widetilde{\mu}|$, therefore, $\overline{\mu\mu'} > \widetilde{\mu}$. The result follows immediately. $\square$

Let $R(-n) = 0$ be a relation among homogeneous operators of degree $-n/6$. Assume that $R(-n) \in \bar{\mathscr{U}}^{(s)}$. These relations typically come from the generating function identities presented in Chapter 5. We will apply the relation $R(-n) = 0$ on a vector of the form $v = X(-\mu_*)v_0$, where $\mu_* \in \mathscr{P}$.

In what follows we will describe the vectors $R(-n)X(-\mu_*)v_0$ modulo the subspace $V_{(\widetilde{\mu})}$ (Recall Notation 4.4.3). We think of $\mu_0$ as the lowest term that we want to keep track in our calculation.

*Remark.* Notice that if

$$\mu_1 \prec \mu_2 \prec \dots \mu_k$$

are lowest $k$ partitions of $\mathscr{P}_s(n)$, then

$$\overline{\mu_1\mu_*} \prec \overline{\mu_2\mu_*} \prec \cdots \prec \overline{\mu_k\mu_*}$$

have the same relative order in $\mathscr{P}_{s+l(\mu_*)}(n + |\mu_*|)$ (see Lemma 4.4.4).

In general, any operator $R(-n) \in \bar{\mathscr{U}}^{(s)}$ can be expressed as a sum of standard monomials classified into three categories:

(A) the terms having no Heisenberg element—terms of the form $X(-\mu)$, $\mu \in \mathscr{Q}(n)$, $l(\mu) \leq s$.

(B) the terms containing negative Heisenberg element(s)—terms of the form

$$\alpha(-\lambda_1)X(-\mu)\alpha(\lambda_2), \tag{4.4.16}$$

$\mu \in \mathscr{Q}$, $\lambda_1 \neq \varnothing, \lambda_2 \in \mathscr{O}$, such that $|\mu| + |\lambda_1| - |\lambda_2| = n$ and $l(\mu) \leq s$; and

(C) the terms having no negative Heisenberg element, but having some positive Heisenberg element(s)—terms of the form

$$X(-\mu)\alpha(\lambda_2), \tag{4.4.17}$$

$\mu \in \mathscr{Q}$, $\lambda_2 \neq \varnothing \in \mathscr{O}$, such that $|\mu| - |\lambda_2| = n$ and $l(\mu) \leq s$.

**Notation 4.4.9.** Let $R(-n) \in \bar{\mathcal{U}}^{(s)}$ be a homogeneous operator of degree $-n/6$ on $V$. We will write $R(-n)$ as

$$R(-n) \equiv A(-n) + B(-n) + C(-n) \mod \bar{\mathcal{U}}^{(s-1)}, \tag{4.4.18}$$

where

$$A(-n) = \sum_{\mu \in \mathscr{Q}_s(n)} a_\mu \, X(-\mu) \tag{4.4.19}$$

is the sum of all terms of type (A) modulo $\bar{\mathcal{U}}^{(s-1)}$

$$B(-n) = \sum_{\substack{\lambda_1, \lambda_2 \in \mathscr{O} \\ \lambda_1 \neq \varnothing \\ \mu \in \mathscr{Q}_s(n - |\lambda_1| + |\lambda_2|)}} b_{\lambda_1, \mu, \lambda_2} \, \alpha(-\lambda_1) X(-\mu) \alpha(\lambda_2) \tag{4.4.20}$$

is the sum of all terms of type (B) modulo $\bar{\mathcal{U}}^{(s-1)}$, and

$$C(-n) = \sum_{\substack{\lambda \in \mathscr{O}, \, \lambda \neq \varnothing \\ \mu \in \mathscr{Q}_s(n + |\lambda|)}} c_{\mu, \lambda} \, X(-\mu) \alpha(\lambda) \tag{4.4.21}$$

is the sum of all terms of type (C) modulo $\bar{\mathcal{U}}^{(s-1)}$.

Now we will analyze which terms in $R(-n)$ are nontrivial when applied on a vector of the form $v = X(-\mu_*)v_0$ modulo $V_{(\widetilde{\mu})}$ (see Notation 4.4.3).

**Proposition 4.4.10.** *Recall Notation 4.4.3 and Notation 4.4.9. We have*

$$\begin{aligned} A(-n)X(-\mu_*)v_0 &\equiv \sum_{\substack{\mu \in \mathscr{P}_s(n) \\ \mu \preceq \mu_0}} a_\mu \, X(-\mu) X(-\mu_*) v_0 \\ &\equiv \sum_{\substack{\mu \in \mathscr{P}_s(n) \\ \mu \preceq \mu_0}} a_\mu \, X(-\overline{\mu\mu_*}) v_0 \mod V_{(\widetilde{\mu})}. \end{aligned} \tag{4.4.22}$$

*Proof.* We apply each term in $A(-n)$ on $v = X(-\mu_*)v_0$. The result follows from Lemma 4.4.5, Lemma 4.4.6 and Lemma 4.2.4. $\square$

**Proposition 4.4.11.** *Recall Notation 4.4.3 and Notation 4.4.9. We have*

$$B(-n)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})}. \tag{4.4.23}$$

*Proof.* The result follows immediately from Lemma 4.4.7. $\square$

**Proposition 4.4.12.** *Recall Notation 4.4.3 and Notation 4.4.9. We have*

$$C(-n)X(-\mu_*)v_0 \equiv \sum_{\substack{\lambda\in\mathscr{O},\,\lambda\neq\varnothing\\ \mu\in\mathscr{P}_s(n+\lambda)\\ \mu\prec\mu_0}} c_{\mu,\lambda}X(-\mu)\alpha(\lambda)X(-\mu_*)v_0 \mod V_{(\widetilde{\mu})}. \tag{4.4.24}$$

*Proof.* The result follows immediately from Lemma 4.4.5 and Lemma 4.4.8. □

*Remark.* We will see in Chapter 6 that to show that a partition $\mu_0$ is forbidden we will need

$$C(-n)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})}.$$

In what follows, we will describe which terms in $C(-n)$ can be ignored (or under what conditions way may ignore $C(-n)$ entirely), modulo $V_{(\widetilde{\mu})}$. But, first, we need the following notations.

**Notation 4.4.13.** For $\mu_0 \in \mathscr{P}_s(n)$ and $k \in \mathbb{N}$, let

$$\mathscr{S}^{\mu_0} = \left\{ \mu \in \mathscr{P} \,\middle|\, l(\mu) = l(\mu_0),\, |\mu| > |\mu_0|,\, \mu \prec \mu_0 \right\},$$
$$\mathscr{S}^{\mu_0}_k = \left\{ \mu \in \mathscr{S}^{\mu_0} \,\middle|\, |\mu| = |\mu_0| + k \right\}.$$

**Notation 4.4.14.** Let , $\mu_*, \mu_0$ and $\widetilde{\mu}$ as defined before. Then define

$$C^{\mu_0}(-n) = \sum_{\substack{\lambda\in\mathscr{O},\,\lambda\neq\varnothing\\ \mu\in\mathscr{S}^{\mu_0}_{|\lambda|}}} c_{\mu,\lambda}X(-\mu)\alpha(\lambda),$$

$$C^{\mu_0}_k(-n) = \sum_{\substack{\lambda\in\mathscr{O}_k,\,\lambda\neq\varnothing\\ \mu\in\mathscr{S}^{\mu_0}_k}} c_{\mu,\lambda}X(-\mu)\alpha(\lambda),$$

Below, we note a few obvious, but noteworthy facts.

(i) The set $\mathscr{S}^{\mu_0}$ is finite, and $\mathscr{S}^{\mu_0}_k = \emptyset$ for $k$ sufficiently large.

(ii) Note that Proposition 4.4.12 can be written as

$$C(-n)X(-\mu_*)v_0 \equiv C^{\mu_0}(-n)X(-\mu_*)v_0 \mod V_{(\widetilde{\mu})}.$$

Also note that

$$C^{\mu_0}(-n) = \sum_{k>0} C^{\mu_0}_k(-n),$$

and $C^{\mu_0}_k(-n) = 0$ for $k$ sufficiently large (follows from (i)).

(iii) If $\mathscr{S}^{\mu_0} = \emptyset$, then $C^{\mu_0}(-n) = 0$. In this case $C(-n)$ can be ignored, in the sense that $C(-n)X(-\mu_*)v_0 \equiv 0 \mod V_{(\widetilde{\mu})}$.

Assume that $\mathscr{S}^{\mu_0} \neq \emptyset$. Let $k$ be the largest integer such that $\mathscr{S}_k^{\mu_0} \neq \emptyset$. Then $\mathscr{S}_j^{\mu_0} = \emptyset$, for all $j > k$.

**Lemma 4.4.15.** *Let $k$ be as described above. If every $\mu \in \mathscr{S}_k^{\mu_0}$ is reducible by partitions greater than $\mu_0$, then the terms in $C_k^{\mu_0}(-n)$ can be ignored, in the sense that*

$$C_k^{\mu_0}(-n)X(-\mu_*)v_0 \equiv 0 \mod V_{(\widetilde{\mu})}.$$

*Proof.* Let $\mu \in \mathscr{S}_k^{\mu_0}$. Let $\lambda \in \mathscr{O}(k)$, such that $X(-\mu)\alpha(\lambda)$ is a term in $C_k^{\mu_0}(-n)$. Since $\mu$ is reducible by partitions greater than $\mu_0$, we can write

$$X(-\mu) \equiv A' + B' + C' \mod \mathcal{U}^{(s-1)}, \tag{4.4.25}$$

where

$$A' = \sum_{\substack{\mu' \in \mathscr{Q}_s(n+k) \\ \mu' \succ \mu_0}} a'_{\mu'} X(-\mu'), \tag{4.4.26}$$

$$B' = \sum_{\substack{\lambda'_1, \lambda'_2 \in \mathscr{O} \\ \lambda'_1 \neq \varnothing \\ \mu' \in \mathscr{Q}_s(n+k-|\lambda'_1|+|\lambda'_2|)}} b'_{\lambda'_1,\mu',\lambda'_2} \alpha(-\lambda'_1) X(-\mu')\alpha(\lambda'_2), \tag{4.4.27}$$

$$C' = \sum_{\substack{\lambda' \in \mathscr{O}, \lambda' \neq \varnothing \\ \mu' \in \mathscr{Q}_s(n+k+|\lambda'|)}} c'_{\mu',\lambda'} X(-\mu')\alpha(\lambda') \tag{4.4.28}$$

Now

$$\begin{aligned} X(-\mu)\alpha(\lambda)X(-\mu_*)v_0 = {} & A'\alpha(\lambda)X(-\mu_*)v_0 \\ & + B'\alpha(\lambda)X(-\mu_*)v_0 \\ & + C'\alpha(\lambda)X(-\mu_*)v_0. \end{aligned} \tag{4.4.29}$$

By Proposition 4.4.12 and hypothesis, we have

$$A'\alpha(\lambda)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})},$$

and by Proposition 4.4.11, we have

$$B'\alpha(\lambda)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})}.$$

A typical term in $C'X(-\mu_*)v_0$ is of the form

$$X(-\mu')\alpha(\lambda')\alpha(\lambda)X(-\mu_*)v_0, \tag{4.4.30}$$

where $\lambda' \in \mathscr{O}(k')$, $k' \neq 0$, and $\mu' \in \mathscr{P}_s(n+k+k')$. But according to our hypothesis, $\mathscr{S}_{k+k'}^{\mu_0} = \emptyset$. Therefore, $\mu' \succ \mu_0$. Thus,

$$X(-\mu')\alpha(\lambda')\alpha(\lambda)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})},$$

by Lemma 4.4.8 as required. □

We now generalize the above lemma. Recall $k$ is the largest integer with $\mathscr{S}_k^{\mu_0} \neq \emptyset$.

**Lemma 4.4.16.** *Assume that $\mathscr{S}^{\mu_0} \neq \emptyset$ (Recall Notation 4.4.13). Let $k \in \mathbb{N}$ be the largest such that $\mathscr{S}_k^{\mu_0} \neq \emptyset$. Let $1 \leq j \leq k$. If every $\mu \in \mathscr{S}_i^{\mu_0}$, $i \geq j$, is reducible by partitions greater than $\mu_0$, then the terms in $C_j^{\mu_0}(-n)$ can be ignored, in the sense that*

$$C_j^{\mu_0}(-n)X(-\mu_*)v_0 \equiv 0 \mod V_{(\widetilde{\mu})}.$$

*Proof.* We will use backward finite induction on $j$. We have already proved the statement for $j = k$ in Lemma 4.4.15. Assume $j < k$, and that the result is true for all $k \geq j' > j$.

The induction step is exactly the same as in the proof of Lemma 4.4.15, except for the last step, where we use the induction hypothesis.

Let $\mu \in \mathscr{S}_j^{\mu_0}$. Let $\lambda \in \mathscr{O}(j)$, such that $X(-\mu)\alpha(\lambda)$ is a term in $C_j^{\mu_0}(-n)$. Since $\mu$ is reducible by partitions greater than $\mu_0$, we can write

$$X(-\mu) \equiv A' + B' + C' \mod \mathcal{U}^{(s-1)}, \tag{4.4.31}$$

where

$$A' = \sum_{\substack{\mu' \in \mathscr{Q}_s(n+j) \\ \mu' \succ \mu_0}} a'_{\mu'} X(-\mu'), \tag{4.4.32}$$

$$B' = \sum_{\substack{\lambda'_1, \lambda'_2 \in \mathscr{O} \\ \lambda'_1 \neq \varnothing \\ \mu' \in \mathscr{Q}_s(n+j-|\lambda'_1|+|\lambda'_2|)}} b'_{\lambda'_1,\mu',\lambda'_2} \alpha(-\lambda'_1) X(-\mu') \alpha(\lambda'_2), \tag{4.4.33}$$

$$C' = \sum_{\substack{\lambda' \in \mathscr{O}, \lambda' \neq \varnothing \\ \mu' \in \mathscr{Q}_s(n+j+|\lambda'|)}} c'_{\mu',\lambda'} X(-\mu') \alpha(\lambda') \tag{4.4.34}$$

Now

$$\begin{aligned} X(-\mu)\alpha(\lambda)X(-\mu_*)v_0 = {} & A'\alpha(\lambda)X(-\mu_*)v_0 \\ & + B'\alpha(\lambda)X(-\mu_*)v_0 \\ & + C'\alpha(\lambda)X(-\mu_*)v_0. \end{aligned} \tag{4.4.35}$$

By Proposition 4.4.12 and hypothesis, we have

$$A'\alpha(\lambda)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})},$$

and by Proposition 4.4.11, we have

$$B'\alpha(\lambda)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})}.$$

A typical term in $C'X(-\mu_*)v_0$ is of the form

$$X(-\mu')\alpha(\lambda')\alpha(\lambda)X(-\mu_*)v_0, \tag{4.4.36}$$

where $\lambda' \in \mathscr{O}(j')$, $j' \neq 0$, and $\mu' \in \mathscr{P}_s(n+j+j')$. If $\mu' \succ \mu_0$ then we are done. Assume that $\mu' \in \mathscr{S}^{\mu_0}_{j+j'}$. But then by the induction hypothesis,

$$X(-\mu')\alpha(\lambda')\alpha(\lambda)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})},$$

as required. □

We summarize our results below.

**Proposition 4.4.17.** *Recall Notation 4.4.3, Notation 4.4.9, Notation 4.4.13 and Notation 4.4.14. Then we have*

$$\begin{aligned} R(-n)X(-\mu_*)v_0 \equiv {} & \sum_{\substack{\mu\in\mathscr{P}_s(n)\\ \mu\preceq\mu_0}} a_\mu X(-\overline{\mu\mu_*})v_0 \\ & + \sum_{\substack{\lambda\in\mathscr{O},\,\lambda\neq\varnothing\\ \mu\in\mathscr{P}_s(n+|\lambda|)\\ \mu\prec\mu_0}} c_{\mu,\lambda} X(-\mu)\alpha(\lambda)X(-\mu_*)v_0 \mod V_{(\widetilde{\mu})}, \end{aligned} \tag{4.4.37}$$

*where $a_\mu, c_{\mu,\lambda}$ are constants from the definition of $R(-n)$, as defined in Notation 4.4.9.*

*In particular, if either*

(i) $\mathscr{S}^{\mu_0} = \emptyset$*, i.e., there is no $\mu \prec \mu_0$ with $|\mu| > n$, or*

(ii) *every* $\mu \in \mathscr{S}^{\mu_0}$ *(i.e.,* $\mu \in \mathscr{P}_s$ *with* $\mu \prec \mu_0$ *and* $|\mu| > n$*), is reducible by partitions greater than* $\mu_0$*,*

*then*

$$C(-n)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})}, \tag{4.4.38}$$

*and therefore,*

$$R(-n)X(-\mu_*)v_0 \equiv \sum_{\substack{\mu\in\mathscr{P}_s(n)\\ \mu\preceq\mu_0}} a_\mu X(-\overline{\mu\mu_*})v_0 \mod V_{(\widetilde{\mu})}. \tag{4.4.39}$$

*Proof.* The first assertion follows directly from Lemma 4.4.5, Proposition 4.4.10, Proposition 4.4.11 and Proposition 4.4.12.

For the second assertion, if $\mathscr{S}^{\mu_0} = \emptyset$ then the statement is obvious, since

$$C^{\mu_0}(-n) = 0.$$

If every $\mu \in \mathscr{S}^{\mu_0}$ is reducible by partitions greater than $\mu_0$, then

$$C(-n)X(-\mu_*)v_0 \in V_{(\widetilde{\mu})}.$$

by Lemma 4.4.16 (with $j = 1$). $\square$

*Remark.* The above proposition shows that the only standard monomials in $R(-n)$ (along with the coefficients) that we need to keep track when we apply $R(-n)$ to the vector $X(-\mu_*)v_0$ modulo the subspace $V_{(\widetilde{\mu})}$, where $\mu_* \in \mathscr{P}$ is arbitrary, are of the form $X(-\mu)$ ($\mu \in \mathscr{P}_s(n)$, $\mu \preceq \mu_0$), and $X(-\mu)\alpha(\lambda)$ ($\mu \in \mathscr{P}_s(n+|\lambda|)$, $\mu \prec \mu_0$). We will be using this argument extensively in Chapter 6.

*Remark.* Sometimes, we will create new relations by multiplying $R(-n)$ on the left by a suitable $X(-\mu')$. Notice that the standard monomials of type (A), and the terms of type (C) remain of the same type when we do this multiplication. However, after straightening out, some of the terms in $X(-\mu')B(-n)$ may yield terms of the type (A), in addition to terms of the type (B).

# Chapter 5

# Generating Function Identities

We will use (4.3.2) as the starting point for the spanning set for any level 4 standard module. To reduce the set further, we need relations among the elements in this set. These relations come from certain generating function identities.

In § 5.1, we define the "product" generating functions $X(\alpha_1, \ldots, \alpha_s; z) \in \mathcal{U}^{(s)}[[z^{\pm 1/6}]]$ for $s \in \mathbb{N}$, $\alpha_1, \cdots, \alpha_s \in L$. These generating functions are intuitively thought of as the "product" of $X(\alpha_1; z), \ldots, X(\alpha_s; z)$. Obviously, the product $X(\alpha_1; z) \cdots X(\alpha_s; z)$ does not exist, as the individual factors are doubly infinite series in $z^{\pm 1/6}$.

In § 5.2, we describe the action of these product generating functions on the basic module $U$. We derive a few identities by "conjugating" the generating function $X(\beta; z)$ with the exponential generating functions $E^{\pm}(\alpha; z)$. These calculations are to be viewed in $(\operatorname{End} U)[[z^{\pm 1/6}]]$, via the representation $\mathcal{U} \to \operatorname{End} U$.

In § 5.3, we work out the action of the "product" generating functions on the tensor product module, $U^{\otimes 4}$. We derive the identities by "conjugating" (note that this is not a conjugation in the strict sense) these generating functions by $E^{\pm}(\alpha; z)$. Since each of the level 4 standard modules are sub-modules of $U^{\otimes 4}$, these identities are valid for any level 4 standard module $V$.

For each $n \in \mathbb{Z}$, the coefficient of $z^{-n/6}$ in a product generating function is a homogeneous operator of degree $n/6$ in the image of $\mathcal{U}$ inside $\operatorname{End} V$, via the representation $\mathcal{U} \to \operatorname{End} V$. This is called the homogeneous component of degree $n/6$ of the said product generating function. We can express them in terms of the standard monomials (4.2.7) using various straightening lemmas described in § 4.2, as necessary. In § 5.4, we compute the coefficients of monomials of the form $X(-\mu)$, $\mu \in \mathscr{Q}$, in this expression for those product generating functions that are involved in the identities presented in § 5.3.

For the first two sections, the details in more generality can be found in [Cap88, Cap92]. The computations in § 5.3 are in similar spirit as [Cap92].

## 5.1 Definitions: The Product Generating Functions

All calculations in this section are done inside $\mathcal{U}(\widetilde{\mathfrak{g}})[[z^{\pm 1/6}]]$. Recall from Chapter 2, $\mathcal{U}(\widetilde{\mathfrak{g}})$ can be thought of embedded in $\operatorname{End} S$, via the isomorphism Proposition 2.3.1.

Recall from (2.2.37),

$$X(\alpha; z) = 6^{-\langle\alpha,\alpha\rangle/2}\sigma(\alpha)E^-(-\alpha; z)E^+(-\alpha; z), \tag{5.1.1}$$

for $\alpha \in L$, where (recall from (2.2.23))

$$\sigma(\alpha) = 2^{-\langle\alpha,\alpha\rangle/2}(1-\omega^{-1})^{\langle\nu\alpha,\alpha\rangle}(1-\omega^{-2})^{\langle\nu^2\alpha,\alpha\rangle}. \tag{5.1.2}$$

In particular, $\sigma(0) = 1$, and $X(0; z) = 1$ on $\mathcal{U}(\widetilde{\mathfrak{g}})$. For $\alpha \in L_2$, $\sigma(\alpha) = \frac{\omega_0\sqrt{3}}{6}$. By abuse of notation, we will denote $\sigma = \frac{\omega_0\sqrt{3}}{6}$ for convenience.

For $\alpha, \beta \in L_2$, define

$$I_n(\alpha,\beta) = \left\{ p \in \mathbb{Z}_6 \,\middle|\, \langle\nu^p\alpha,\beta\rangle = n \right\}, \tag{5.1.3a}$$

$$I_\pm(\alpha,\beta) = \left\{ p \in \mathbb{Z}_6 \,\middle|\, \pm\langle\nu^p\alpha,\beta\rangle > 0 \right\}. \tag{5.1.3b}$$

Notice that $I(n) = I_n(\alpha, \beta)$, in the notation of Proposition 2.2.6. Also recall the following polynomials in $x = \left(\frac{z_2}{z_1}\right)^{1/6}$:

$$P_0[\alpha,\beta](x) = \prod_{p\in I_-(\alpha,\beta)} (1-\omega^{-p}x)^{-\langle\nu^p\alpha,\beta\rangle}, \tag{5.1.4a}$$

$$Q_0[\alpha,\beta](x) = \prod_{p\in I_+(\alpha,\beta)} (1-\omega^{-p}x)^{\langle\nu^p\alpha,\beta\rangle}, \tag{5.1.4b}$$

as used in Proposition 2.2.3 and (2.2.14). Recalling that $\langle\alpha,\beta\rangle = \pm1, \pm2$ for $\alpha, \beta \in L_2$, we can further simplify $P_0$ and $Q_0$ as

$$P_0[\alpha,\beta](x) = \prod_{p\in I_{-1}(\alpha,\beta)}(1-\omega^{-p}x)\prod_{p\in I_{-2}(\alpha,\beta)}(1-w^{-p}x)^2, \tag{5.1.5a}$$

$$Q_0[\alpha,\beta](x) = \prod_{p\in I_1(\alpha,\beta)}(1-\omega^{-p}x)\prod_{p\in I_2(\alpha,\beta)}(1-w^{-p}x)^2. \tag{5.1.5b}$$

Recall the notion of limit as defined in Definition 2.1.4. For simplicity, we are going to use the following convention:

**Notation 5.1.1.** Let $Z(z_1, \ldots, z_s) \in A[[z_1^{\pm 1/6}, \ldots, z_s^{\pm 1/6}]]$ be any expression (where $A$ is any algebra over $\mathbb{C}$). We will use the following notation

$$\lim Z(z_1, \ldots, z_s) = \lim_{z_1^{1/6}, \cdots, z_s^{1/6} \to z^{1/6}} Z(z_1, \ldots, z_s),$$

for abbreviation.

**Notation 5.1.2.** Throughout, we will use the abbreviation

$$x = \left(\frac{z_2}{z_1}\right)^{1/6},$$

or more generally,

$$x_{ij} = \left(\frac{z_j}{z_i}\right)^{1/6}, \quad \text{for } i < j,$$

unless otherwise mentioned.

**Proposition 5.1.3.** *For $\alpha, \beta \in L_2$, there exists a Laurent polynomial $P(x) = P[\alpha, \beta](x)$ with constant coefficients such that the limit*

$$\lim P(x) X(\alpha; z_1) X(\beta; z_2)$$

*exists.*

*Proof.* Using Proposition 2.1.6 it enough to find $P(x)$ such that

$$P(x)[X(\alpha; z_1), X(\beta; z_2)] = 0. \tag{5.1.6}$$

From Proposition 2.2.6, we see that each term in the expansion of $[X(\alpha; z_1), X(\beta; z_2)]$ contains one of the following factors: $\delta(\omega^{-p}x)$ for $p \in I_{-1}(\alpha, \beta)$, $\mathrm{D}\,\delta(\omega^{-p}x)$ for $p \in I_{-2}(\alpha, \beta)$, or $\delta(\omega^{-p}x)$ for $p \in I_{-2}(\alpha, \beta)$. Note that for $\alpha, \beta \in L_2$, $\langle \alpha, \beta \rangle$ can only assume the values $\pm 1, \pm 2$.

Using Proposition 2.1.7 and (5.1.5), we see that

$$\begin{aligned} P_0[\alpha, \beta](x) \delta(\omega^{-p}x) = 0 \qquad & \text{for } p \in I_{-1}(\alpha, \beta), \\ P_0[\alpha, \beta](x)\, \mathrm{D}\, \delta(\omega^{-p}x) = 0 \qquad & \text{for } p \in I_{-2}(\alpha, \beta). \end{aligned} \tag{5.1.7}$$

Therefore, we can take $P(x)$ to be any multiple of $P_0(x)$ by a Laurent polynomial. $\square$

*Remark.* Sometimes we may want $\overline{P} = \lim P(x) \neq 0$. This is only possible if $P_0[\alpha, \beta](x)$ doesn't have any factor of $(1-x)$, which is the case when $\langle \alpha, \beta \rangle > 0$.

*Remark.* Also, from the above proof, it is obvious that $P(x)$ only depends on the angle between $\alpha$ and $\beta$. We will denote the choice of $P(x)$ by $P_{\langle \alpha, \beta \rangle}(x)$, depending on $\alpha, \beta \in L_2$.

**Definition 5.1.4.** For $\alpha, \beta \in L_2$, and $P$ as in Proposition 5.1.3. Then we define

$$X(\alpha, \beta; z) = X_P(\alpha, \beta; z) = \lim P(x) X(\alpha, z_1) X(\beta, z_2).$$

*Remark.* We will drop the subscript $P$, if the choice of $P$ is not important and there is no danger of confusion.

*Remark.* In view of Proposition 5.1.3, the order of $\alpha$ and $\beta$ in $X(\alpha, \beta; z)$ does not matter.

The product generating function can be generalized for more than two factors.

**Definition 5.1.5.** Let $\alpha_i, \ldots, \alpha_s \in L_2$. Denote by

$$x_{ij} = \left( \frac{z_j}{z_i} \right)^{1/6},$$

for $1 \leq i < j \leq s$. Let

$$P_{ij} = P_{\langle \alpha_i, \alpha_j \rangle}(x_{ij}),$$

and

$$P = \prod_{1 \leq i < j \leq s} P_{ij}.$$

Then we define the product generating function

$$X(\alpha_1, \ldots, \alpha_s; z) = X(\alpha_1, \ldots, \alpha_s; z) = \lim P X(\alpha_1, z_1) \cdots X(\alpha_s, z_s).$$

**Notation 5.1.6.** If $\alpha_1 = \cdots = \alpha_s = \alpha$, then denote $X_P(\alpha_1, \ldots, \alpha_s; z)$ by $X^{(s)}(\alpha; z)$.

*Remark.* Notice that the product generating function $X(\alpha_1, \ldots, \alpha_s; z) \in \mathcal{U}^{(s)}$.

## 5.2 Identities on the Basic Module

In this section, we describe the action of the product generating functions on the basic module, $U$. The generating functions $X(\alpha; z)$, $E^{\pm}(\alpha; z)$, and the product generating functions $X(\alpha_1, \ldots, \alpha_s; z)$ are thought of as generating functions with coefficients in $\operatorname{End} U$.

**Proposition 5.2.1.** *Let $\alpha, \beta \in L_2$. Then on the basic module $U$ we have*

$$X(\alpha, \beta; z) = 6^{\langle \alpha, \beta \rangle} \frac{\sigma(\alpha)\sigma(\beta)}{\sigma(\alpha+\beta)} A X(\alpha+\beta; z),$$

*where*

$$A = \lim \frac{Q_0}{P_0} P,$$

*$P_0 = P_0[\alpha, \beta](x)$, $Q_0 = Q_0[\alpha, \beta](x)$, and $P = P_{\langle \alpha, \beta \rangle}(x)$.*

*Proof.* First, notice that $A$ exists because $P$ is a multiple of $P_0$. Using (5.1.1) and Proposition 2.2.3, we have

$$\begin{aligned} X(\alpha; z_1) X(\beta; z_2) &= \frac{\sigma(\alpha)\sigma(\beta)}{6^{(\langle \alpha, \alpha \rangle + \langle \beta, \beta \rangle)/2}} E^-(-\alpha; z_1) E^+(-\alpha; z_1) E^-(-\beta; z_2) E^+(-\beta; z_2) \\ &= \frac{\sigma(\alpha)\sigma(\beta)}{6^{(\langle \alpha, \alpha \rangle + \langle \beta, \beta \rangle)/2}} \frac{Q_0}{P_0} E^-(-\alpha; z_1) E^-(-\beta; z_2) E^+(-\alpha; z_1) E^+(-\beta; z_2) \\ &= 6^{\langle \alpha, \beta \rangle} \frac{\sigma(\alpha)\sigma(\beta)}{\sigma(\alpha+\beta)} \frac{Q_0}{P_0} 6^{-\langle \alpha+\beta, \alpha+\beta \rangle/2} \sigma(\alpha+\beta) \\ &\quad \times E^-(-\alpha; z_1) E^-(-\beta; z_2) E^+(-\alpha; z_1) E^+(-\beta; z_2) \end{aligned}$$

Now multiply both sides by $P$, and take the limit to get

$$X(\alpha, \beta; z) = 6^{\langle \alpha, \beta \rangle} \frac{\sigma(\alpha)\sigma(\beta)}{\sigma(\alpha+\beta)} A X(\alpha+\beta; z). \qquad \square$$

**Corollary 5.2.2.** *Let $\alpha, \beta \in L_2$. If $\langle \alpha, \beta \rangle > 0$, then $X(\alpha, \beta; z) = 0$ on the basic module $U$. In particular, $X^{(2)}(\alpha; z) = 0$ on $U$.*

*Proof.* If $\langle \alpha, \beta \rangle > 0$, then $Q_0$ contains a factor of $(1-x)$. Therefore $\lim Q_0 = 0$, and hence $A = 0$. $\square$

*Remark.* If $\alpha, \beta \in L_2$, with $\langle \alpha, \beta \rangle = -1$, then $\alpha + \beta \in L_2$.

We state the generalizations of Proposition 5.2.1 and Corollary 5.2.2, which follow easily from similar arguments.

**Proposition 5.2.3.** *Let $\alpha_1, \ldots, \alpha_s \in L_2$. Then on the basic module $U$ we have*

$$X(\alpha_1, \ldots, \alpha_s; z) = \prod_{1 \le i < j \le s} 6^{\langle \alpha_i, \alpha_j \rangle} \frac{\sigma(\alpha_i)\sigma(\alpha_j)}{\sigma(\alpha_i + \alpha_j)} A_{ij} X(\alpha_1 + \cdots + \alpha_s; z),$$

*where*

$$A_{ij} = \lim \frac{Q_0[\alpha_i, \alpha_j]}{P_0[\alpha_i, \alpha_j]} P_{\langle \alpha_i, \alpha_j \rangle}.$$

**Corollary 5.2.4.** *Let $\alpha_1, \dots, \alpha_s \in L_2$. If $\langle \alpha_i, \alpha_j \rangle < 0$ for some $1 \le i < j \le s$, then $X(\alpha_1, \dots, \alpha_s; z) = 0$ on the basic module $U$.*

We need the following generating function identities on the basic module to discover the identities on the standard modules of level 4. The following identity is to be thought of as the result of "conjugating" $X(\beta; z)$ by the exponentials $E^{\pm}(\alpha; z)$.

**Proposition 5.2.5.** *Let $\alpha, \beta \in L$. On the basic module $U$, we have*

$$E^-(\alpha; z) X(\beta; z) E^+(\alpha; z) = 6^{\langle \alpha, \alpha \rangle/2 - \langle \alpha, \beta \rangle} \frac{\sigma(\beta)}{\sigma(\alpha - \beta)} X(\beta - \alpha; z).$$

*Proof.* Using (5.1.1), we have

$$\begin{aligned} E^-(\alpha; z) X(\beta; z) E^+(\alpha; z) &= 6^{-\langle \beta, \beta \rangle/2} \sigma(\beta) E^-(\alpha; z) E^-(-\beta; z) E^+(-\beta; z) E^+(\alpha; z) \\ &= \frac{6^{-\langle \beta, \beta \rangle/2}}{6^{-\langle \alpha-\beta, \alpha-\beta \rangle/2}} \cdot \frac{\sigma(\beta)}{\sigma(\alpha - \beta)} \\ &\quad \times \left( 6^{-\langle \alpha-\beta, \alpha-\beta \rangle/2} \sigma(\alpha - \beta) E^-(\alpha - \beta; z) E^+(\alpha - \beta; z) \right) \\ &= 6^{\langle \alpha, \alpha \rangle/2 - \langle \alpha, \beta \rangle} \frac{\sigma(\beta)}{\sigma(\alpha - \beta)} X(\beta - \alpha; z), \end{aligned}$$

since $\sigma(-\alpha) = \sigma(\alpha)$. □

**Notation 5.2.6.** Since $\sigma(\alpha)$ only depends on the length of $\alpha$, we will denote, by abuse of notation, $\sigma = \sigma(\alpha) = \frac{\omega_0 \sqrt{3}}{6}$ for any $\alpha \in L_2$ (see (2.2.24)).

**Corollary 5.2.7.** *Let $\alpha \in L_2$. On the basic module $U$, we have*

(i) $E^-(\alpha; z) E^+(\alpha; z) = \frac{6}{\sigma} X(-\alpha; z)$,

(ii) $E^-(\alpha; z) X(\alpha; z) E^+(\alpha; z) = \frac{\sigma}{6} \mathbf{1}$,

(iii) $E^-(\alpha; z) X(\nu^{\pm 1} \alpha; z) E^+(\alpha; z) = X(\nu^{\pm 2} \alpha; z)$,

*where $\mathbf{1}$ is the identity operator on $U$.*

*Proof.* (i) Take $\beta = 0$ in Proposition 5.2.5. Notice that $\sigma(0) = 1$, and $X(0; z) = \mathbf{1}$ on $U$.

(ii) Take $\beta = \alpha$ in Proposition 5.2.5.

(iii) Take $\beta = \nu^{\pm 1} \alpha$ in Proposition 5.2.5. □

## 5.3 Identities on the Level 4 Standard Modules

In this section, we will derive the generating function identities similar to Proposition 5.2.5 and Corollary 5.2.7 on any level 4 standard module $V$. By Proposition 3.1.6 any level 4 standard module is contained inside the tensor product of four copies of the basic module. Therefore, these identities can be viewed in $\operatorname{End} U^{\otimes 4}$. Conceptually, these identities can be thought of as the result of "conjugating" (not in the strict sense) various product generating functions $X(\beta_1, \ldots, \beta_s; z)$ with the exponentials $E^{\pm}(\alpha; z)$. These computations are done in the same spirit as [Cap92].

Let $U$ be the basic module, and $U^{\otimes 4} = U \otimes U \otimes U \otimes U$ be the tensor product module. Since the components of $X(\alpha; z)$, for $\alpha \in L_2$, are elements of the Lie algebra $\widetilde{\mathfrak{g}}$, $X(\alpha; z)$ acts on $U^{\otimes 4}$ as a primitive element:

$$\begin{aligned} X(\alpha; z) = X(\alpha; z) \otimes 1 \otimes 1 \otimes 1 + 1 \otimes X(\alpha; z) \otimes 1 \otimes 1 \\ + 1 \otimes 1 \otimes X(\alpha; z) \otimes 1 + 1 \otimes 1 \otimes 1 \otimes X(\alpha; z), \end{aligned} \tag{5.3.1}$$

where $X(\alpha; z)$ is viewed as an operator on $U^{\otimes 4}$ on the RHS, and as an operator on $U$ on the LHS.

On the other hand, the exponential generating function $E^{\pm}(\alpha; z)$, for $\alpha \in L$, acts on $U^{\otimes 4}$ as a group-like element:

$$E^{\pm}(\alpha; z) = E^{\pm}(\alpha; z) \otimes E^{\pm}(\alpha; z) \otimes E^{\pm}(\alpha; z) \otimes E^{\pm}(\alpha; z), \tag{5.3.2}$$

where $E^{\pm}(\alpha; z)$ is viewed as an operator on $U^{\otimes 4}$ on the LHS, and as an operator on $U$ on the RHS.

The identities presented in this section, are obtained by "conjugating" various product generating functions $X(\alpha_1, \ldots, \alpha_s; z)$ by $E^{\pm}(\alpha; z)$. In the presentation below, we will classify the results by $s$, i.e., by the highest number of factors in the product generating function occurring in the identity. We will call this number as the degree of the identity. We will present one degree 2 identity and four degree 3 identities on $U^{\otimes 4}$. We start with the degree 2 identity.

**Proposition 5.3.1.** *Let $\alpha \in L_2$. Then on $U^{\otimes 4}$ we have*

$$E^{-}(\alpha; z) X^{(2)}(\alpha; z) E^{+}(\alpha; z) = X^{(2)}(-\alpha; z).$$

*Proof.* The polynomial $P$ in the definition of $X^{(2)}(\alpha;z)$ and $X^{(2)}(-\alpha;z)$ are the same, since $\langle\alpha,\alpha\rangle = \langle-\alpha,-\alpha\rangle$. By Corollary 5.2.2, if both $X(\alpha;z)$s (or $X(-\alpha;z)$s) act on the same tensorand of $U^{\otimes 4}$ then it yields 0. Therefore, on $U^{\otimes 4}$,

$$X^{(2)}(\alpha;z) = 2\overline{P}\left(\underbrace{X(\alpha;z)\otimes X(\alpha;z)\otimes 1\otimes 1+\cdots}_{\text{total 6 similar terms}}\right),$$

because there are $\binom{4}{2}$ possible way to distribute the two $X(\alpha;z)$s in 4 tensorands without having two of them acting on the same tensorand, and we get two copies for each term having $X(\alpha;z)$s acting on similar positions. Similarly,

$$X^{(2)}(-\alpha;z) = 2\overline{P}\left(\underbrace{X(-\alpha;z)\otimes X(-\alpha;z)\otimes 1\otimes 1+\cdots}_{\text{total 6 similar terms}}\right).$$

By "conjugating" $X^{(2)}(\alpha;z)$ by $E^{\pm}(\alpha;z)$ on $U^{\otimes 4}$, we get

$$\begin{aligned}
\text{LHS} = 2\overline{P}\Big(&E^-(\alpha;z)X(\alpha;z)E^+(\alpha;z)\otimes E^-(\alpha;z)X(\alpha;z)E^+(\alpha;z)\\
&\otimes E^-(\alpha;z)E^+(\alpha;z)\otimes E^-(\alpha;z)E^+(\alpha;z)\\
&+\cdots(5\text{ similar terms})\Big)\\
= 2\overline{P}\Big(&1\otimes 1\otimes X(-\alpha;z)\otimes X(-\alpha;z)+\cdots(5\text{ similar terms})\Big),
\end{aligned}$$

using Corollary 5.2.7. This is precisely the same as the RHS. □

We present the four degree 3 identities below.

**Proposition 5.3.2.** *Let $\alpha \in L_2$. Then on $U^{\otimes 4}$ we have*

$$E^-(\alpha;z)X^{(3)}(\alpha;z)E^+(\alpha;z) = 6\overline{P}\left(\frac{\sigma}{6}\right)^2 X(-\alpha;z).$$

*Proof.* Let $P$ be the polynomial in the definition of $X^{(3)}(\alpha;z)$. Then, on $U^{\otimes 4}$, we have

$$X^{(3)}(\alpha;z) = 6\overline{P}\left(\underbrace{X(\alpha;z)\otimes X(\alpha;z)\otimes X(\alpha;z)\otimes 1+\cdots}_{\text{total 4 similar terms}}\right).$$

Now "conjugating" $X^{(3)}(\alpha;z)$ with $E^{\pm}(\alpha;z)$ on $U^{\otimes 4}$, we get

$$\begin{aligned}
\text{LHS} = 6\overline{P}\Big(&E^{-}(\alpha;z)X(\alpha;z)E^{+}(\alpha;z) \otimes E^{-}(\alpha;z)X(\alpha;z)E^{+}(\alpha;z) \\
&\otimes E^{-}(\alpha;z)X(\alpha;z)E^{+}(\alpha;z) \otimes E^{-}(\alpha;z)E^{+}(\alpha;z) \\
&+ \cdots (\text{3 similar terms})\Big) \\
= \left(\frac{\sigma}{6}\right)^2 & 6\overline{P}\Big(1 \otimes 1 \otimes 1 \otimes X(-\alpha;z) + \cdots (\text{3 similar terms})\Big) \\
= 6\overline{P} & \left(\frac{\sigma}{6}\right)^2 X(-\alpha;z) = \text{RHS},
\end{aligned}$$

using Corollary 5.2.7. □

**Proposition 5.3.3.** *Let $\alpha \in L_2$. Then on $U^{\otimes 4}$ we have*

$$E^{-}(\alpha;z)X(\alpha,\alpha,\nu\alpha;z)E^{+}(\alpha;z) = \left(\frac{\sigma}{3}\overline{P}_1\overline{P}_2\right) X(-\alpha,\nu^2\alpha;z).$$

*Proof.* Let $P_{\langle\alpha,\beta\rangle}$ denote the polynomial so that $P_{\langle\alpha,\beta\rangle}[X(\alpha;z_1),X(\beta;z_2)] = 0$. Then we can use $P_L = P_2(x_{12})P_1(x_{13})P_1(x_{23})$, in the definition of $X(\alpha,\alpha,\nu\alpha;z)$, and $P_R = P_1(x)$ in the definition of $X(-\alpha,\nu^2\alpha;z)$.

Since, each of the pairwise inner products of the roots among $\alpha,\alpha,\nu\alpha$ are positive, only terms where each of $X(\alpha;z)$ and $X(\nu\alpha;z)$ acts on distinct tensorands of $U^{\otimes 4}$ survive (by Proposition 5.2.1), when acting on $U^{\otimes 4}$. Therefore we have

$$X(\alpha,\alpha,\nu\alpha;z) = 2\overline{P_1}^2\overline{P_2}\left(\underbrace{X(\alpha;z)\otimes X(\alpha;z)\otimes X(\nu\alpha;z)\otimes 1 + \cdots}_{\text{total 12 similar terms}}\right),$$

and

$$X(-\alpha,\nu^2\alpha;z) = \overline{P_1}\left(\underbrace{X(-\alpha;z)\otimes X(\nu^2\alpha;z)\otimes 1\otimes 1 + \cdots}_{\text{total 12 similar terms}}\right),$$

on $U^{\otimes 4}$.

"Conjugating" $X(\alpha,\alpha,\nu\alpha;z)$ by $E^{\pm}(\alpha;z)$, we get

$$\begin{aligned}
\text{LHS} = 2\overline{P_1}^2\overline{P_2}\Big(&E^{-}(\alpha;z)X(\alpha;z)E^{+}(\alpha;z) \otimes E^{-}(\alpha;z)X(\alpha;z)E^{+}(\alpha;z) \\
&\otimes E^{-}(\alpha;z)X(\nu\alpha;z)E^{+}(\alpha;z) \otimes E^{-}(\alpha;z)E^{+}(\alpha;z) \\
&+ \cdots (\text{11 similar terms})\Big) \\
= \left(\frac{\sigma}{6}\right) & 2\overline{P_1}^2\overline{P_2}\Big(1 \otimes 1 \otimes X(\nu^2\alpha;z) \otimes X(-\alpha;z) + \cdots (\text{11 similar terms})\Big) \\
= \left(\frac{\sigma}{3}\overline{P_1P_2}\right) & X(-\alpha,\nu^2\alpha;z) = \text{RHS},
\end{aligned}$$

using Corollary 5.2.7. $\square$

**Proposition 5.3.4.** *Let $\alpha \in L_2$. Then on $U^{\otimes 4}$ we have*

$$E^-(\alpha; z)X(\alpha, \nu\alpha, \nu\alpha; z)E^+(\alpha; z) = X(-\alpha, \nu^2\alpha, \nu^2\alpha; z).$$

*Proof.* Using the same notation as before, let $P = P_1(x_{12})P_1(x_{13})P_2(x_{23})$ in the definition of both $X(\alpha, \nu\alpha, \nu\alpha; z)$ and $X(-\alpha, \nu^2\alpha, \nu^2\alpha; z)$.

Then on $U^{\otimes 4}$, we have

$$X(\alpha, \nu\alpha, \nu\alpha; z) = 2\overline{P}\left(\underbrace{X(\alpha; z) \otimes X(\nu\alpha; z) \otimes X(\nu\alpha; z) \otimes 1 + \cdots}_{\text{total 12 similar terms}}\right),$$

and

$$X(-\alpha, \nu^2\alpha, \nu^2\alpha; z) = 2\overline{P}\left(\underbrace{X(-\alpha; z) \otimes X(\nu^2\alpha; z) \otimes X(\nu^2\alpha; z) \otimes 1 + \cdots}_{\text{total 12 similar terms}}\right).$$

"Conjugating" $X(\alpha, \nu\alpha, \nu\alpha; z)$ by $E^{\pm}(\alpha; z)$, we get

$$\begin{aligned}
\text{LHS} = 2\overline{P}\Big(&E^-(\alpha; z)X(\alpha; z)E^+(\alpha; z) \otimes E^-(\alpha; z)X(\nu\alpha; z)E^+(\alpha; z) \\
&\otimes E^-(\alpha; z)X(\nu\alpha; z)E^+(\alpha; z) \otimes E^-(\alpha; z)E^+(\alpha; z) \\
&+ \cdots (\text{11 similar terms})\Big) \\
= 2\overline{P}&\Big(1 \otimes X(\nu^2\alpha; z) \otimes X(\nu^2\alpha; z) \otimes X(-\alpha; z) + \cdots (\text{11 similar terms})\Big) \\
= X(&-\alpha, \nu^2\alpha, \nu^2\alpha; z) = \text{RHS},
\end{aligned}$$

using Corollary 5.2.7. $\square$

**Proposition 5.3.5.** *Let $\alpha \in L_2$. Then on $U^{\otimes 4}$ we have*

$$E^-(\alpha; z)X(\alpha, \nu\alpha, \nu^{-1}\alpha; z)E^+(\alpha; z) = X(-\alpha, \nu^2\alpha, \nu^{-2}\alpha; z).$$

*Proof.* Using the same notation as before, let $P = P_1(x_{12})P_1(x_{13})P_{-1}(x_{23})$ in the definition of both $X(\alpha, \nu\alpha, \nu^{-1}\alpha; z)$ and $X(-\alpha, \nu^2\alpha, \nu^{-2}\alpha; z)$.

Let $P_0 = P_0[\nu\alpha, \nu^{-1}\alpha]$ and $Q_0 = Q_0[\nu\alpha, \nu^{-1}\alpha]$. Using Proposition 5.2.1, we have

$$\begin{aligned}
X(\nu\alpha, \nu^{-1}\alpha; z) &= \frac{\sigma}{6}AX(\alpha; z), \\
X(\nu^2\alpha, \nu^{-2}\alpha; z) &= \frac{\sigma}{6}AX(-\alpha; z),
\end{aligned}$$

where $A = \overline{P}_{-1}\frac{\overline{Q}_0}{\overline{P}_0}$ is a constant as defined in Proposition 5.2.1. Therefore, both $X(\nu\alpha; z)$ and $X(\nu^{-1}\alpha; z)$ may act on the same tensorand of $U^{\otimes 4}$. The same is true for $X(\nu^2\alpha; z)$ and $X(\nu^{-2}\alpha; z)$.

Thus on $U^{\otimes 4}$, we have, letting $B = \lim P_1(x_{12})P_1(x_{13})P_{-1}(x_{23})\frac{Q_0(x_{23})}{P_0(x_{23})}$,

$$\begin{aligned} X(\alpha, \nu\alpha, \nu^{-1}\alpha; z) = \overline{P}&\Bigg(\underbrace{X(\alpha; z) \otimes X(\nu\alpha; z) \otimes X(\nu^{-1}\alpha; z) \otimes 1 + \cdots}_{\text{total 24 similar terms}}\Bigg) \\ &+ 2B\Bigg(\underbrace{X(\alpha; z) \otimes X(\alpha; z) \otimes 1 \otimes 1}_{\text{total 6 similar terms}}\Bigg), \end{aligned}$$

and

$$\begin{aligned} X(-\alpha, \nu^2\alpha, \nu^{-2}\alpha; z) = \overline{P}&\Bigg(\underbrace{X(-\alpha; z) \otimes X(\nu^2\alpha; z) \otimes X(\nu^{-2}\alpha; z) \otimes 1 + \cdots}_{\text{total 24 similar terms}}\Bigg) \\ &+ 2B\Bigg(\underbrace{X(-\alpha; z) \otimes X(-\alpha; z) \otimes 1 \otimes 1}_{\text{total 6 similar terms}}\Bigg). \end{aligned}$$

The result follows from "conjugating" $X(\alpha, \nu\alpha, \nu^{-1}\alpha; z)$ by $E^{\pm}(\alpha; z)$, and using Corollary 5.2.7. ☐

*Remark.* In the above proof, $\overline{P} = 0$, but $B \neq 0$.

*Remark.* Since, we are free to multiply the polynomials $P$ by any scalar, and since $\overline{P}_1, \overline{P}_2 \neq 0$, we will assume without loss of generality that $\overline{P_1} = \overline{P_2} = 1$. This will make the computation of the coefficients more convenient. Similarly, we will scale $P_{-1}$ in such a way that makes $B = 1$ in Proposition 5.3.5.

## 5.4 Coefficients

In this section, we compute the coefficient of $X(-\mu)$ in various product generating functions which are involved in the degree 2 or 3 identities presented in § 5.3.

The coefficient of $z^{-n/6}$ in $X(\alpha_1, \ldots, \alpha_s; z) \in \mathcal{U}^{(s)}[[z^{\pm n/6}]]$ gives the homogeneous component of degree $\frac{n}{6}$. For $A_2$, any $\beta \in L_2$, can be expressed as $\nu^k\alpha$ (where $\alpha \in L_2$ is our chosen fixed root). By (2.2.39), $X(\beta; m) = \omega^{km}X(m)$. Thus, the homogeneous component of degree $\frac{n}{6}$ consists of terms of the form $X(-\mu)$, where $\mu \in \mathbb{Z}^s$, with $|\mu| = n$. Of course, the parts of $\mu$ could be out of order — we need to use Lemma 4.2.4

to straighten out $X(-\mu)$ in order to express it in terms of the standard monomials. From Lemma 4.2.4, it follows that $X(-\mu) \in X(-\bar{\mu}) + \mathcal{U}^{(s-1)}$, i.e., $X(-\mu) \equiv X(-\bar{\mu})$ mod $\mathcal{U}^{(s-1)}$. ($\bar{\mu} \in \mathscr{Q}$ is the result of rearranging $\mu$ in non-ascending order).

**Lemma 5.4.1.** *Let $\mu = (p, q) \in \mathscr{Q}_2(n)$.*

(i) *The coefficient of $X(-\mu)$ in $X^{(2)}(\alpha; z)$ expressed in terms of the standard monomials is*

$$c_a^{(2)}(\mu) = \begin{cases} 1 & \text{if } p = q, \\ 2 & \text{if } p \neq q. \end{cases} \tag{5.4.1}$$

(ii) *The coefficient of $X(-\mu)$ in $X^{(2)}(-\alpha; z)$ expressed in terms of the standard monomials is*

$$c_b^{(2)}(\mu) = \begin{cases} 1 & \text{if } p = q, \\ 2(-1)^{p+q} & \text{if } p \neq q. \end{cases} \tag{5.4.2}$$

*Proof.* (i) If $p = q$ then there is only one way to get the term $X(-p)X(-q)$. If $p \neq q$ then there are two terms $X(-p)X(-q)$ and $X(-q)X(-p)$ which corresponds to $X(-\mu)$ after possible straightening out by Lemma 4.2.4. The result follows from the definition of $X^{(2)}(\alpha; z)$.

(ii) The proof is similar to (i), keeping in mind that $X(-\alpha; -p) = (-1)^p X(\alpha; -p)$.

$\square$

*Remark.* The coefficient of $X(-\mu)$, $\mu = (p, q) \in \mathscr{Q}$ in $X^{(2)}(\alpha; z)$ can be easily computed as the coefficient of $x^2$, if $p = q$; or the coefficient of $xy$, if $p \neq q$, in the polynomial $(x+y)^2$ in two commuting variables. Similarly, the coefficient of $X(-\mu)$ in $X^{(2)}(-\alpha; z)$ is the coefficient of $x^2$, if $p = q$; or the coefficient of $xy$ in the polynomial $((-1)^p x + (-1)^q y)^2$. We used this method to compute the coefficients in the Maple worksheet (see Appendix A).

**Definition 5.4.2.** We call a function, $f\colon \mathscr{Q}_k \to \mathbb{C}$ periodic with periodicity $m$ (or, $m$-periodic, for brevity), if $f(\mu) = f(\mu + \mathbf{m})$ for all $\mu \in \mathscr{Q}_k$, i.e., if $f$ is invariant under increasing/decreasing each part by $m$.

**Corollary 5.4.3.** *$c_a^{(2)}$ and $c_b^{(2)}$ are 1-periodic functions on $\mathscr{Q}_2$.*

*Proof.* It is obvious from the formulae in Lemma 5.4.1. □

**Notation 5.4.4.** For $\mu \in \mathscr{Q}$, we denote by $\#\mu$ the number of distinct parts in $\mu$.

**Lemma 5.4.5.** *For $\mu = (p, q, r) \in \mathscr{Q}_3(n)$, let $a$ be the most frequent part in $\mu$, $b$ be the next most frequent part (if any), and $c$ be the least frequent part (if any). If $p, q, r$ are all distinct, we may simply take $a = p$, $b = q$ and $c = r$. If two of $p, q, r$ are the same, then take $a$ to be the one that is repeated twice, and $b$ the unique one. If $p = q = r$, take $a = p$.*

(i) *The coefficient of $X(-\mu)$ in $X^{(3)}(\alpha; z)$ expressed in terms of the standard monomials is given by*

$$c_1^{(3)}(\mu) = \begin{cases} 1 & \text{if } \#\mu = 1, \\ 3 & \text{if } \#\mu = 2, \\ 6 & \text{if } \#\mu = 3. \end{cases} \tag{5.4.3}$$

(ii) *The coefficient of $X(-\mu)$ in $X(\alpha, \alpha, \nu\alpha; z)$ expressed in terms of the standard monomials is given by*

$$c_2^{(3)}(\mu) = \begin{cases} \omega^{-a} & \text{if } \#\mu = 1, \\ 2\omega^{-a} + \omega^{-b} & \text{if } \#\mu = 2, \\ 2(\omega^{-a} + \omega^{-b} + \omega^{-c}) & \text{if } \#\mu = 3. \end{cases} \tag{5.4.4}$$

(iii) *The coefficient of $X(-\mu)$ in $X(\alpha, \nu\alpha, \nu\alpha; z)$ expressed in terms of the standard monomials is given by*

$$c_{3a}^{(3)}(\mu) = \begin{cases} \omega^{-2a} & \text{if } \#\mu = 1, \\ \omega^{-2a} + 2\omega^{-a-b} & \text{if } \#\mu = 2, \\ 2(\omega^{-a-b} + \omega^{-b-c} + \omega^{-c-a}) & \text{if } \#\mu = 3. \end{cases} \tag{5.4.5}$$

(iv) *The coefficient of $X(-\mu)$ in $X(-\alpha, \nu^2\alpha, \nu^2\alpha; z)$ expressed in terms of the standard monomials is given by*

$$c_{3b}^{(3)}(\mu) = \begin{cases} \omega^{-7a} & \text{if } \#\mu = 1, \\ \omega^{-4a-3b} + 2\omega^{-5a-2b} & \text{if } \#\mu = 2, \\ 2(\omega^{-3a-2b-2c} + \omega^{-2a-3b-2c} + \omega^{-2a-2b-3c}) & \text{if } \#\mu = 3. \end{cases} \tag{5.4.6}$$

(v) *The coefficient of $X(-\mu)$ in $X(\alpha, \nu\alpha, \nu^{-1}\alpha; z)$ expressed in terms of the standard monomials is given by*

$$c_{4a}^{(3)}(\mu) = \begin{cases} \omega^{-6a} = 1 & \text{if } \#\mu = 1, \\ \omega^{-6a} + \omega^{-5a-b} + \omega^{-a-5b} & \text{if } \#\mu = 2, \\ \big(\omega^{-5a-b} + \omega^{-a-5b} + \omega^{-5b-c} & \\ \quad + \omega^{-b-5c} + \omega^{-5c-a} + \omega^{-c-5a}\big) & \text{if } \#\mu = 3. \end{cases} \tag{5.4.7}$$

(vi) *The coefficient of $X(-\mu)$ in $X(-\alpha, \nu^2\alpha, \nu^{-2}\alpha; z)$ expressed in terms of the standard monomials is given by*

$$c_{4b}^{(3)}(\mu) = \begin{cases} \omega^{-9a} = (-1)^{-a} & \text{if } \#\mu = 1, \\ \omega^{-6a-3b} + \omega^{-5a-4b} + \omega^{-7a-2b} & \text{if } \#\mu = 2, \\ \big(\omega^{-3a-2b-4c} + \omega^{-3a-4b-2c} & \\ \quad + \omega^{-4a-3b-2c} + \omega^{-2a-3b-4c} & \text{if } \#\mu = 3. \\ \quad + \omega^{-2a-4b-3c} + \omega^{-4a-2b-3c}\big) & \end{cases} \tag{5.4.8}$$

*Proof.* The proof is similar to the proof of Lemma 5.4.1. Let $a, b, c$ be as defined above, with the following convention that $b$ and $c$ be 0 when they are not defined (i.e., if $\#\mu = 1$ or 2). We note that the coefficient of $X(-\mu)$ in $X(\nu^i\alpha, \nu^j\alpha, \nu^k\alpha; z)$ can be extracted as the coefficient of $x^3$, $x^2y$ or $xyz$ in the polynomial

$$\left(\omega^{-ia}x + \omega^{-ib}y + \omega^{-ic}z\right)\left(\omega^{-ja}x + \omega^{-jb}y + \omega^{-jc}z\right)\left(\omega^{-ka}x + \omega^{-kb}y + \omega^{-kc}z\right)$$

depending on whether $\#\mu$ is 1, 2 or 3, respectively. □

*Remark.* Let $C\colon \mathscr{Q}_3 \to \mathbb{C}$ be any one of the above coefficient functions (viz., $c_1^{(3)}$, $c_2^{(3)}$, $c_{3a}^{(3)}$, $c_{3b}^{(3)}$, $c_{4a}^{(3)}$ or $c_{4c}^{(3)}$). Then each term in $C(\mu)$ can be expressed as a power of $\omega$. The exponents of $\omega$ are linear in $a, b, c$, of the form $m_1a + m_2b + m_3c$. Note that the sum $m = m_1 + m_2 + m_3$ is constant for all terms and all cases (i.e., $\#\mu = 1, 2$ or 3). Therefore, if we increase each part in $\mu$ by 1, the exponent increases by $m$, giving us a factor of $\omega^m$. Thus we have,

$$C(\mu + \mathbf{1}) = \omega^m C(\mu).$$

The following periodicity properties follow from this observation.

**Corollary 5.4.6.** *Let $\mu \in \mathscr{Q}_3$.*

(i) $c_1^{(3)}$ *has periodicity 1:*

$$c_1^{(3)}(\mu + \mathbf{1}) = c_1^{(3)}(\mu).$$

(ii) $c_2^{(3)}$ *has periodicity 6:*

$$c_2^{(3)}(\mu + \mathbf{1}) = \omega^{-1} c_2^{(3)}(\mu).$$

(iii) $c_{3a}^{(3)}$ *has periodicity 3:*

$$c_{3a}^{(3)}(\mu + \mathbf{1}) = \omega^{-2} c_{3a}^{(3)}(\mu).$$

(iv) $c_{3b}^{(3)}$ *has periodicity 6:*

$$c_{3b}^{(3)}(\mu + \mathbf{1}) = \omega^{-1} c_{3b}^{(3)}(\mu).$$

(v) $c_{4a}^{(3)}$ *has periodicity 1:*

$$c_{4a}^{(3)}(\mu + \mathbf{1}) = c_{4a}^{(3)}(\mu).$$

(vi) $c_{4b}^{(3)}$ *has periodicity 2:*

$$c_{4b}^{(3)}(\mu + \mathbf{1}) = -c_{4b}^{(3)}(\mu).$$

# Chapter 6

# Forbidden Partitions

In this chapter, we discover the forbidden partitions using the relations coming from the degree 2 and degree 3 identities in Chapter 5. The results of this chapter are valid for all level 4 modules. Let $V$ denote a level 4 standard module for $\widetilde{\mathfrak{g}}$ with highest weight $\Lambda$, and a highest weight vector $v_0$.

The generating function identities are of the form $R(z) = 0$, where $R(z) \in \bar{\mathcal{U}}^{(s)}[[z^{\pm 6}]]$ $(s = 2, 3)$, where $\bar{\mathcal{U}}$ is the image of $\mathcal{U}$ in $\operatorname{End} V$. Given any such identity, the homogeneous component of degree $-n/6$ (i.e., the coefficient of $z^{n/6}$ in $R(z)$), denoted $R(-n)$, gives us a relation among homogeneous operators of degree $-n/6$ on $V$:

$$R(-n) = 0.$$

Recall the notations and results from Chapter 4. In particular, we will rely heavily on Proposition 4.4.17. Recall that we call a partition $\mu \in \mathscr{P}$ *reducible* if $X(-\mu)v_0 \in V_{(\mu)}$ (Definition 4.3.6). Proposition 4.4.1 shows that any partition that ends with a reducible partitions is also reducible. By Proposition 4.3.15, we may remove elements of the form $\alpha(\lambda)X(-\mu)v_0$ from the spanning set (4.3.2) if $\mu \in \mathscr{P}$ is reducible, for all $\lambda \in \mathscr{O}$.

Recall the notion of “forbidden” partition from Definition 4.3.8. We call a partition $\mu$ forbidden, if any partition having $\mu$ as a sub-partition (not just suffix) is also forbidden. This is stronger than the result of Proposition 4.4.1. To show that $\mu$ is forbidden, we are going to apply our relation $R(-n) \in \bar{\mathcal{U}}^{(s)}$ on a vector $X(-\mu_*)v_0$ for an arbitrary $\mu_* \in \mathscr{P}$.

The results presented in this chapter are valid for any level 4 standard module, since the generating function identities (see Chapter 5) that we are going to use are valid on $U^{\otimes 4}$.

In the following discussion a "term" refers to a summand of $R(-n)$ expressed as the standard monomial $\alpha(\lambda_1)X(-\mu')\alpha(\lambda_2)$, for some $\lambda_1, \lambda_2 \in \mathscr{O}$, $\mu' \in \mathscr{Q}$, such that $n = |\mu'| + |\lambda_1| - |\lambda_2|$ and $l(\mu') \leq s$. Alternatively, a "term" may also refer to one of the form $E'(-i)X(-\mu')E'(j)$, for some $i, j \geq 0$, $\mu' \in \mathscr{Q}$, $n = |\mu'| + i - j$ and $l(\mu') \leq s$, where $E'(n) = E(-\alpha; n)$. (Since, $E'(\pm i)$ can be expressed as a linear combination of $\alpha(\pm\lambda)$, with $|\lambda| = i$).

Recall from Notation 4.4.9 that the terms in $R(-n)$ can be classified into three categories: type (A) (having no Heisenberg element), type (B) (having a negative Heisenberg element) and type (C) (having a positive Heisenberg element, but no negative Heisenberg element). Proposition 4.4.17 shows that only certain terms in $A(-n)$ and $C(-n)$ are relevant for our calculation. The rest of the terms, therefore, can be ignored (modulo a suitable subspace $V_{(\widetilde{\mu})}$).

In section § 6.1, we show that any partition containing two consecutive integers as parts are forbidden. These are coming from the degree 2 generating function identities in § 5.3.

In section § 6.2, we show that the first three (i.e., the least three with respect to "$<$" on $\mathscr{P}$) triplets of $n$ are forbidden. In addition if $n$ is odd, the fourth triplet is also forbidden. These are coming from the degree 3 identities in § 5.3.

The results of these two sections are analogous to the results of [Cap88, Cap93] for level 2 and 3 modules. We call the partitions shown to be forbidden in these two sections as *regular* forbidden partitions.

The interesting aspect of level 4 module is that there are forbidden partitions of arbitrary length (starting from length 3) following a simple pattern. We call these forbidden partitions as *exceptional* forbidden partitions. These partitions do not arise directly from the generating function identities. The relevant operator relations are obtained by multiplying $R(-n)$ (coming from the degree 2 generating function identity) by a suitable $X(-\mu_L)$ on the left, and/or a suitable $X(-\mu_R)$ on the right.

In § 6.3, we discuss the exceptional forbidden triplets. These are of the form $(k + 4, k, k)$, for $k$ odd, i.e., satisfying difference condition (see Definition 4.1.9) $[4-, 0]$. The partition $(5, 1, 1)$ also satisfy the same difference condition, however, it is also the

4-th partition of the odd integer 7, and it can be thought of a regular forbidden triplet.

In § 6.4, we discuss the longer exceptional forbidden partitions. These partitions satisfy the difference condition (see Definition 4.1.9 $[3-, 2^*, 3, 0]$, where $2^*$ denote zero or more occurrence of 2. Examples of such partitions include

$$(9, 6, 3, 3), (11, 8, 6, 3, 3), (13, 10, 8, 6, 3, 3), \dots .$$

Notice that if the first part is not odd then the partition ends with a regular forbidden triplet of difference condition $[3-, 0]$ (the 4-th triplet of an odd integer). Therefore, either way, any partition containing a sub-partition satisfying the difference condition $[3, 2^*, 3, 0]$ is forbidden.

In § 6.5, we summarize the results of this chapter in one place, to be quoted later for convenience. We also add some observations about the "periodicity properties" of the forbidden partitions presented in this chapter.

Recall the notations and the tools and techniques described in § 4.4.

## 6.1 Forbidden Pairs

In this section, we prove that pairs (i.e., partition into two parts) of the form $(k+1, k)$ (i.e., satisfying the difference condition [1]) are forbidden. Thus, any vector of the form $\alpha(\lambda)X(-\mu)v_0$, $\lambda \in \mathscr{O}$, $\mu \in \mathscr{P}$ such that $\mu$ contains two consecutive integers as parts, can be removed from the spanning set (4.3.2). These results come from the degree 2 identity Proposition 5.3.1.

Consider the following generating function in $(\operatorname{End} V)[[z^{\pm 1/6}]]$

$$R^{(2)}(z) = X^{(2)}(\alpha; z) - E^-(-\alpha; z)X^{(2)}X(-\alpha; z)E^+(-\alpha; z). \tag{6.1.1}$$

Then the identity Proposition 5.3.1 can be expressed as

$$R^{(2)}(z) = 0. \tag{6.1.2}$$

Let $R^{(2)}(-n) \in \bar{\mathcal{U}}^{(2)}$ denote the homogeneous component of $R^{(2)}(z)$ of degree ${-n}/{6}$. Then

$$R^{(2)}(z) = \sum_{\mathfrak{n} \in \mathbb{Z}} R^{(2)}(-n) z^{n/6}. \tag{6.1.3}$$

Therefore, we have

$$R^{(2)}(-n) = 0 \in \operatorname{End} V, \tag{6.1.4}$$

for all $n$.

Recall the coefficient functions $c_a^{(2)}$ and $c_b^{(2)}$ from Lemma 5.4.1. Let $c^{(2)} \colon \mathscr{Q}_2 \to \mathbb{C}$ be the function given by

$$c^{(2)}(\mu) = c_a^{(2)}(\mu) - c_b^{(2)}(\mu), \tag{6.1.5}$$

for all $\mu \in \mathscr{Q}_2$. From Lemma 5.4.1, we see that

$$c^{(2)}\left(\mu_1^{(2)}(n)\right) = \begin{cases} 0 & \text{if } n \text{ is even,} \\ 4 & \text{if } n \text{ is odd.} \end{cases} \tag{6.1.6}$$

Let $E'(n) = E(-\alpha; n)$, for all $n \in \mathbb{Z}$. Notice that, for $n > 0$, $E'(\pm n)$ is a linear combination of elements of the form $\alpha(\pm\lambda)$, $\lambda \in \mathscr{O}(n)$.

Also recall the notations $A(-n), B(-n), C(-n)$ for a given $R(-n)$ from Notation 4.4.9. In this case, we have (modulo $\bar{\mathcal{U}}^{(1)}$)

$$A^{(2)}(-n) = \sum_{\mu \in \mathscr{Q}_2(n)} c^{(2)}(\mu) X(-\mu), \tag{6.1.7a}$$

$$B^{(2)}(-n) = -\sum_{\substack{i>0,\, j\geq 0 \\ \mu \in \mathscr{Q}_2(n-i+j)}} c_b^{(2)}(\mu) E'(-i) X(-\mu) E'(j), \tag{6.1.7b}$$

$$C^{(2)}(-n) = -\sum_{\substack{j>0 \\ \mu \in \mathscr{Q}_2(n+j)}} c_b^{(2)}(\mu) X(-\mu) E'(j). \tag{6.1.7c}$$

Thus, we have

$$R^{(2)}(-n) \equiv A^{(2)}(-n) + B^{(2)}(-n) + C^{(2)}(-n) \mod \bar{\mathcal{U}}^{(1)}, \tag{6.1.8}$$

for all $n$.

Let $\mu_1^{(2)}(n)$ denote the least partition in $\mathscr{P}_2(n)$. Then

$$\mu_1^{(2)}(n) = \begin{cases} (k,k) & \text{if } n = 2k \ (k \geq 1), \\ (k+1,k) & \text{if } n = 2k+1 \ (k \geq 1). \end{cases} \tag{6.1.9}$$

Note that $\mu_1^{(2)}(n+2) = \mu_1^{(2)}(n) + \mathbf{1}$ (see Definition 4.1.12).

We take $\mu_0 = \mu_1^{(2)}(n)$, in the context of § 4.4 as in Notation 4.4.3. In the notation of Notation 4.4.13, we have

$$\mathscr{S}^{\mu_0} = \emptyset, \tag{6.1.10}$$

**Proposition 6.1.1.** *Partitions of the form* $(k+1, k)$*, i.e., partitions containing difference condition* [1] *are forbidden.*

*Proof.* Let $\mu_0 = \mu_1^{(2)}(2k+1) = (k+1, k)$ for some $k > 0$. Let $\mu_* \in \mathscr{P}$ be arbitrary and $\widetilde{\mu} = \overline{\mu_0 \mu_*}$ (as in Notation 4.4.3). We want to prove that $\widetilde{\mu}$ is reducible, i.e., $X(-\widetilde{\mu})v_0 \in V_{(\widetilde{\mu})}$.

Since $R^{(2)}(z) = 0$, we have, in particular,

$$R^{(2)}(-2k-1) = 0.$$

By Proposition 4.4.17, since $\mathscr{S}^{\mu_0} = \emptyset$, we have

$$0 = R(-2k-1)X(-\mu_*)v_0 \equiv 4X(-\overline{\mu_0\mu_*})v_0 = 4X(-\widetilde{\mu})v_0 \mod V_{(\widetilde{\mu})}.$$

Therefore, $X(-\widetilde{\mu})v_0 \in V_{(\widetilde{\mu})}$, as required. $\square$

## 6.2 Regular Forbidden Triplets

In this section, we prove that the first three triplets (i.e., partitions into three parts) of any $n > 0$ are forbidden. In addition, if $n$ is odd, then the fourth triplet is also forbidden. We will prove this in two steps. First we show that these partitions are reducible. Then we will use this result to show that these partitions are, in fact, forbidden using the Proposition 4.4.17. These forbidden triplets are "regular" in the sense that they follow directly from the the four degree 3 generating function identities Proposition 5.3.2, Proposition 5.3.3, Proposition 5.3.4 and Proposition 5.3.5.

Let $\mu_i^{(3)}(n)$ denote the $i$-th smallest partition of $n$ into three parts. We list the first four partitions below for $n$ sufficiently large.

$$\left.\begin{aligned}\mu_1^{(3)}(n) &= (k, k, k)\\ \mu_2^{(3)}(n) &= (k+1, k, k-1)\\ \mu_3^{(3)}(n) &= (k+1, k+1, k-2)\\ \mu_4^{(3)}(n) &= (k+2, k-1, k-1)\end{aligned}\right\} \text{ if } n = 3k,\ k > 2, \tag{6.2.1a}$$

$$\left.\begin{aligned}\mu_1^{(3)}(n) &= (k+1, k, k)\\ \mu_2^{(3)}(n) &= (k+1, k+1, k-1)\\ \mu_3^{(3)}(n) &= (k+2, k, k-1)\\ \mu_4^{(3)}(n) &= (k+2, k+1, k-2)\end{aligned}\right\} \text{ if } n = 3k+1,\ k > 2, \tag{6.2.1b}$$

$$\left.\begin{aligned}\mu_1^{(3)}(n) &= (k+1, k+1, k)\\ \mu_2^{(3)}(n) &= (k+2, k, k)\\ \mu_3^{(3)}(n) &= (k+2, k+1, k-1)\\ \mu_4^{(3)}(n) &= (k+2, k+2, k-2)\end{aligned}\right\} \text{ if } n = 3k+2,\ k > 2. \tag{6.2.1c}$$

The above pattern for first four triplets holds for $n \geq 9$. We list below the the triplets for $n < 9$.

$$\mathscr{P}_3(3) = \{(1,1,1)\} \tag{6.2.2a}$$

$$\mathscr{P}_3(4) = \{(2,1,1)\} \tag{6.2.2b}$$

$$\mathscr{P}_3(5) = \{(2,2,1),(3,1,1)\} \tag{6.2.2c}$$

$$\mathscr{P}_3(6) = \{(2,2,2),(3,2,1),(4,1,1)\} \tag{6.2.2d}$$

$$\mathscr{P}_3(7) = \{(3,2,2),(3,3,1),(4,2,1),(5,1,1)\} \tag{6.2.2e}$$

$$\mathscr{P}_3(8) = \{(3,3,2),(4,2,2),(4,3,1),(5,2,1)\} \tag{6.2.2f}$$

Consider the following generating functions in $(\operatorname{End} V)[[z^{\pm 1/6}]]$.

$$R_1^{(3)}(z) = X^{(3)}(\alpha; z), \tag{6.2.3a}$$

$$R_2^{(3)}(z) = X(\alpha, \alpha, \nu\alpha; z), \tag{6.2.3b}$$

$$R_3^{(3)}(z) = X(\alpha, \nu\alpha, \nu\alpha; z) - E^-(-\alpha; z)X(-\alpha, \nu^2\alpha, \nu^2\alpha; z)E^+(-\alpha; z), \tag{6.2.3c}$$

$$R_4^{(3)}(z) = X(\alpha, \nu\alpha, \nu^{-1}\alpha; z) - E^{-1}(-\alpha; z)X(-\alpha, \nu^2\alpha, \nu^{-2}\alpha; z)E^+(-\alpha; z). \tag{6.2.3d}$$

Using Proposition 5.3.2, Proposition 5.3.3, Proposition 5.3.4 and Proposition 5.3.5, we have

$$R_i^{(3)}(z) = 0 \in (\operatorname{End} V)[[z^{\pm 1/6}]], \tag{6.2.4}$$

for $1 \leq i \leq 4$. Let $R_i^{(3)}(-n) \in \bar{\mathcal{U}}^{(3)}$ denote the homogeneous component of $R_i^{(3)}(z)$ of degree ${-n}/{6}$, i.e.,

$$R_i^{(3)}(z) = \sum_{n \in \mathbb{Z}} R_i^{(3)}(-n) z^{n/6}. \tag{6.2.5}$$

Therefore, we have

$$R_i^{(3)}(-n) = 0, \tag{6.2.6}$$

for all $n$.

Recall the coefficient functions $c_1^{(3)}$, $c_2^{(3)}$, $c_{3a}^{(3)}$, $c_{3b}^{(3)}$, $c_{4a}^{(3)}$ and $c_{4b}^{(3)}$ from Lemma 5.4.5. Define the following functions on $\mathcal{Q}_3$ by

$$c_1^{(3)}(\mu) = c_1^{(3)}(\mu), \tag{6.2.7a}$$

$$c_2^{(3)}(\mu) = c_2^{(3)}(\mu), \tag{6.2.7b}$$

$$c_3^{(3)}(\mu) = c_{3a}^{(3)}(\mu) - c_{3b}^{(3)}(\mu), \tag{6.2.7c}$$

$$c_4^{(3)}(\mu) = c_{4a}^{(3)}(\mu) - c_{4b}^{(3)}(\mu). \tag{6.2.7d}$$

*Remark.* Observe that $c_4^{(3)}(\mu) = 0$ if $|\mu|$ is even.

By Corollary 5.4.6, we have $c_i^{(3)}(\mu) = c_i^{(3)}(\mu + \mathbf{6})$. Since for the first four triplets of $n$, $n \geq 9$, follows the pattern (6.2.1), therefore we have

$$\mu_j^{(3)}(n + 18) = \mu_j^{(3)}(n) + \mathbf{6}, \tag{6.2.8}$$

and

$$c_i^{(3)}\left(\mu_j^{(3)}(n + 18)\right) = c_i^{(3)}\left(\mu_j^{(3)}(n)\right), \tag{6.2.9}$$

for $1 \le i \le 4$, $1 \le j \le 4$ and $n \ge 9$. Also, for uniformity, define the functions

$$\begin{aligned} c_{1b}^{(3)}, c_{2b}^{(3)} \colon \mathscr{Q}_3 &\to \mathbb{C} \\ \mu &\mapsto 0. \end{aligned} \tag{6.2.10}$$

Define $A_i^{(3)}(-n), B_i^{(3)}(-n)$ and $C_i^{(3)}(-n)$ modulo $\bar{\mathscr{U}}^{(2)}$, corresponding to $R_i^{(3)}(-n)$ as in Notation 4.4.9 by

$$A_i^{(3)}(-n) = \sum_{\mu \in \mathscr{Q}_3(n)} c_i^{(3)}(\mu) X(-\mu), \tag{6.2.11a}$$

$$B_i^{(3)}(-n) = -\sum_{\substack{i>0,\, j\ge 0 \\ \mu \in \mathscr{Q}_3(n-i+j)}} c_{ib}^{(3)}(\mu) E'(-i) X(-\mu) E'(j), \tag{6.2.11b}$$

$$C_i^{(3)}(-n) = -\sum_{\substack{j>0 \\ \mu \in \mathscr{Q}_3(n+j)}} c_{ib}^{(3)}(\mu) X(-\mu) E'(j). \tag{6.2.11c}$$

Notice that $B_i^{(3)}(-n) = C_i^{(3)}(-n) = 0$ for $i = 1, 2$. Thus, we have

$$R_i^{(3)}(-n) \equiv A_i^{(3)}(-n) + B_i^{(3)}(-n) + C_i^{(3)}(-n) \mod \bar{\mathscr{U}}^{(2)}, \tag{6.2.12}$$

for all $n$.

**Proposition 6.2.1.** *The smallest three triplets of any $n > 0$ are reducible. If $n$ is odd, then the fourth smallest triplet is also reducible.*

*Proof.* We only need to prove this statement for finitely many $3 \le n \le 26$, because of the periodicity properties as discussed in (6.2.8). Note that $\mathscr{P}_3(n) = \emptyset$ unless $n \ge 3$.

The basic idea is to use all four relations coming from the degree 3 identities. We take $\mu_0 = \mu_k^{(3)}(n)$, where $k = 4$ if $n$ is odd and the partition exist, $k = 4$ if $n$ is even and the partition exist, or the largest $k$ (for $n \le 5$) such that the corresponding partition exist. Let and $\mu_* = \varnothing$ in the context of Notation 4.4.3.

To treat all cases uniformly, we will adopt the convention that $X(-\mu_j^{(3)}(n)) = 0$ if the corresponding partition does not exist for $j \le 4$. (e.g., $\mu_2^{(3)}$ does not exist for $n = 3$). In that case, $\mu_0$ is going to be largest partition that is defined (e.g., $\mu_0 = \mu_1^{(3)}$ for $n = 3$).

Since $R_i^{(3)}(-n) = 0$, we have, using Proposition 4.4.17,

$$0 = R_i^{(3)}(-n) v_0 \equiv \sum_{j=1}^{4} c_i\left(\mu_j^{(3)}(n)\right) X\left(-\mu_j^{(3)}(n)\right) v_0 \mod V_{(\mu_0)}, \tag{6.2.13}$$

for $1 \leq i \leq 4$. We collect the coefficients in the matrix

$$M(n) = \begin{pmatrix} c_1^{(3)}(\mu_1^{(3)}) & \dots & c_1^{(3)}(\mu_4^{(3)}) \\ \vdots & \ddots & \vdots \\ c_4^{(3)}(\mu_1^{(3)}) & \dots & c_4^{(3)}(\mu_4^{(3)}) \end{pmatrix}, \tag{6.2.14}$$

(the last few columns may be absent, if the corresponding partition does not exist). We row-reduce this matrix to $M'(n)$. The computations for a few initial $n$'s are shown in the Maple worksheet attached in Appendix A § A.1.

By the periodicity property (6.2.8), we have

$$M(n+18) = M(n), \tag{6.2.15}$$

for $n \geq 9$. However, the row-reduced matrix $M'(n)$ has stronger periodicity:

$$M'(n+6) = M'(n), \tag{6.2.16}$$

for $n \geq 9$. The computation shows that $M'(n)$ contains a principal identity matrix of rank 4 if $n$ is odd, and of rank 3 if $n$ is even, for $n \geq 7$, with the 1's on the main diagonal. For $n \leq 6$, the rank is same as the number of partitions in $\mathscr{P}_3(n)$. Therefore, we have the desired result. $\square$

*Remark.* Notice that the first two triplets of any $n \geq 3$ follow the same patterns as shown in (6.2.1). Only for $\mu_3^{(3)}(6) = (4,1,1)$ and $\mu_4^{(3)}(7) = (5,1,1)$ fall outside these patterns.

**Proposition 6.2.2.** *The least three triplets of any $n > 0$ are forbidden. If $n$ is odd, then the fourth smallest triplet is also forbidden.*

*Thus, any partition that contains a sub-partition satisfying the difference conditions*

$$[1], [0,0], [0,2], [2,0], [0,3], [3-,0], [0-,4]; \tag{6.2.17}$$

*or that contains $(4,1,1)$ or $(5,1,1)$ as a sub-partition are reducible.*

*Proof.* The second part of the statement is just paraphrasing the first part along with Proposition 6.1.1. (Note that many of the first four triplets contain a forbidden pair).

Let $\mu_* \in \mathscr{P}$ be arbitrary. We need to show that

$$X\left(-\overline{\mu_j^{(3)}(n)\mu_*}\right)v_0 \in V_{\left(\overline{\mu_j^{(3)}(n)\mu_*}\right)}, \tag{6.2.18}$$

where $j \leq 4$ if $n$ is odd, $j \leq 3$ if $n$ is even.

Fix $n \geq 3$. We take $\mu_0 = \mu_k^{(3)}(n)$, where $k = 4$ if $n \geq 7$ is odd, or $k = 3$ if $n \geq 7$ is even, or $k$ is the largest integer such that $\mu_k^{(3)}(n)$ exists if $n \leq 5$. Recall Notation 4.4.3. Then, it is enough to show that

$$X\left(-\overline{\mu_j^{(3)}(n)\mu_*}\right)v_0 \in V_{(\widetilde{\mu})}, \tag{6.2.19}$$

where $j \leq 4$ if $n$ is odd, $j \leq 3$ if $n$ is even.

Recalling Notation 4.4.13, we have

$$\mathscr{S}_1^{\mu_0} = \left\{\mu_1^{(3)}(n+1), \mu_2^{(3)}(n+1)\right\}, \tag{6.2.20a}$$

$$\mathscr{S}_2^{\mu_0} = \left\{\mu_1^{(3)}(n+2)\right\}, \tag{6.2.20b}$$

$$\mathscr{S}_k^{\mu_0} = \emptyset, \qquad \text{if } k > 2. \tag{6.2.20c}$$

(as long as the corresponding partition exists, and omit if it does not). Therefore, by Proposition 6.2.1, we can conclude that all partitions in $\mathscr{S}^{\mu_0}$ are reducible by partitions larger than $\mu_0$. By Proposition 4.4.17, we have

$$0 = R_i^{(3)}(-n)X(-\mu_*)v_0 \equiv \sum_{j=1}^{4} c_i\left(\mu_j^{(3)}(n)\right)X\left(-\overline{\mu_j^{(3)}(n)\mu_*}\right)v_0 \mod V_{(\widetilde{\mu})}, \tag{6.2.21}$$

Now, we proceed the same way as in the proof of Proposition 6.2.1. Notice that the coefficient matrix $M(n)$ is the same as in (6.2.14). And therefore, the conclusion follows similarly as shown in the proof of Proposition 6.2.1. □

*Remark.* We will see in Chapter 7 that any partition ending with $(1,1)$ is reducible in all level 4 standard modules. (We will see that in the $(4,0)$- and $(0,2)$-modules any partition ending with $(1)$ is reducible, and in $(2,1)$-module any partition ending with $(1,1)$ is forbidden). Thus the exceptions $(4,1,1)$ and $(5,1,1)$ to the difference conditions in the statement of the above proposition can be ignored as initial conditions.

## 6.3 Exceptional Forbidden Triplets

In this section, we will prove that triplets of the form $(k+4, k, k)$, where $k \in \mathbb{N}$ is odd, are forbidden, i.e., partitions containing a sub-partition satisfying the difference condition (see Definition 4.1.9) $[4-, 0]$ are reducible. We call them exceptional in the sense that the result does not follow directly from the degree 3 identities alone, but in conjunction with the degree 2 identity. Note that the forbidden triplet $(5, 1, 1)$ also follows the same pattern, but the result follows from the degree 3 identities directly (§ 6.2), and in that sense, it is regular. Therefore, in this section, we only need to prove the result for $k \geq 3$ odd.

We discovered these forbidden triplets experimentally by eliminating the reducible partitions in $\mathscr{P}(n)$ containing any forbidden pairs or triplets from § 6.1 and § 6.2, and then comparing the result with $F_{(4,0)}(n)$ (see Notation 3.2.4) using Corollary 4.3.17, for $n \geq 0$. Let $\mathscr{P}'$ be the result of removing the partitions containing forbidden pairs and triplets. We noticed that the first place where $|\mathscr{P}'(n)| \neq F_{(4,0)}(n)$ was for $n = 13$. In this case, we had an extra partition in $\mathscr{P}'(13)$. We get an extra partition next in $\mathscr{P}'(19)$. The gap of 6 (by the periodicity properties of the forbidden pairs and triplets) suggested that we missed a forbidden triplet. The least triplet left in $\mathscr{P}'(13)$ was the partition $(7, 3, 3)$.

The triplet $(k+4, k, k)$, where $k \geq 3$, is the sixth smallest triplet of $n = 3k+4$. We set $\mu_0 = \mu_6^{(3)}(n)$ (recalling Notation 4.1.19) in the settings of § 4.4 and in Notation 4.4.3. Therefore, in our computation, we only keep track of the first six triplets. We list them below for reference.

$$\begin{aligned}
\mu_1^{(3)}(n) &= (k+2, k+1, k+1), \\
\mu_2^{(3)}(n) &= (k+2, k+2, k), \\
\mu_3^{(3)}(n) &= (k+3, k+1, k), \\
\mu_4^{(3)}(n) &= (k+3, k+2, k-1), \\
\mu_4^{(3)}(n) &= (k+3, k+3, k-1), \\
\mu_4^{(3)}(n) &= (k+4, k, k).
\end{aligned} \tag{6.3.1}$$

We will also need to keep track of first four terms with positive Heisenberg elements.

Therefore we list below the first four triplets of $n+1$.

$$\begin{aligned}
\mu_1^{(3)}(n+1) &= (k+2, k+2, k+1),\\
\mu_2^{(3)}(n+1) &= (k+3, k+1, k+1),\\
\mu_3^{(3)}(n+1) &= (k+3, k+2, k),\\
\mu_4^{(3)}(n+1) &= (k+3, k+3, k-1).
\end{aligned} \tag{6.3.2}$$

Recall the relations given by $R^{(2)}(z) = 0$, $R_1^{(3)}(z) = 0$, $R_2^{(3)}(z) = 0$, $R_3^{(3)}(z) = 0$ and $R_4^{(3)}(z) = 0$ (see (6.1.1), (6.2.3)), and the various coefficient functions from (6.1.5) and (6.2.7).

We get a fifth relation (of degree 3) by multiplying a degree 2 relation by an appropriate $X(\bullet)$ operator on the left. Let

$$R_5^{(3)}(-n) = R_5(-n) = X(-k-3)R^{(2)}(-2k-1). \tag{6.3.3}$$

Also recall the notations $A(-n), B(-n), C(-n)$ for a given $R(-n)$ from Notation 4.4.9. Therefore, $R^{(2)}(-2k-1)$ can be expressed, modulo $\bar{\mathcal{U}}^{(1)}$, as

$$A^{(2)}(-2k-1) = \sum_{\mu \in \mathscr{Q}_2(2k+1)} c^{(2)}(\mu) X(-\mu), \tag{6.3.4a}$$

$$B^{(2)}(-2k-1) = -\sum_{\substack{i>0,\ j\geq 0\\ \mu \in \mathscr{Q}_2(2k+1-i+j)}} c_b^{(2)}(\mu) E'(-i) X(-\mu) E'(j), \tag{6.3.4b}$$

$$C^{(2)}(-2k-1) = -\sum_{\substack{j>0\\ \mu \in \mathscr{Q}_2(2k+1+j)}} c_b^{(2)}(\mu) X(-\mu) E'(j), \tag{6.3.4c}$$

Therefore, we have

$$R^{(2)}(-2k-1) \equiv A^{(2)}(-2k-1) + B^{(2)}(-2k-1) + C^{(2)}(-2k-1) \mod \bar{\mathcal{U}}^{(2)}. \tag{6.3.5}$$

Now, let us look at the terms of $X(-k-3)A^{(2)}(-2k-1)$ after rearranging the $X(\bullet)$ operators using Lemma 4.2.4. In view of Proposition 4.4.17, we only need consider the terms involving $\mu \in \mathscr{P}_3(3k+4)$ such that $\mu \preceq \mu_0 = (k+4, k, k)$. Notice that

$$\overline{(k+3)\cdot\mu_1^{(2)}(2k+1)} = (k+3, k+1, k) \quad = \mu_3^{(3)}(n) \tag{6.3.6a}$$

$$\overline{(k+3)\cdot\mu_2^{(2)}(2k+1)} = (k+3, k+2, k-1) = \mu_4^{(3)}(n) \tag{6.3.6b}$$

$$\overline{(k+3)\cdot\mu_3^{(2)}(2k+1)} = (k+3, k+3, k-2) = \mu_5^{(3)}(n) \tag{6.3.6c}$$

Therefore, modulo $\bar{\mathcal{U}}^{(2)}$, we have

$$X(-k-3)A^{(2)}(-2k-1) = 4X\left(-\mu_3^{(3)}(n)\right) + 4X\left(-\mu_4^{(3)}(n)\right) + 4X\left(-\mu_5^{(3)}(n)\right) + \dots \tag{6.3.7}$$

keeping track of only the relevant terms (using Lemma 5.4.1 for the coefficients).

Let us now look at the terms of $X(-k-3)B^{(3)}(-2k-1)$ after rearranging the $X(\bullet)$ operators by Lemma 4.2.4. In view of Proposition 4.4.11 and Proposition 4.4.17, we will only keep track of the terms that do not have any negative Heisenberg element, and only those that involve $\mu \in \mathscr{P}_3(3k+4)$ such that $\mu \preceq \mu_0 = (k+4, k, k)$. Recall the formula in Corollary 2.2.12. In particular, we have

$$X(-m)E'(-1) = E'(-1)X(-m) - 6X(-m-1), \tag{6.3.8}$$

for any $m \in \mathbb{Z}$.

A typical term in $B^{(2)}(-2k-1)$ is of the form

$$E'(-i)X(-\mu')E'(j) \qquad i > 0,\ j \geq 0, \text{ and } \mu' \in \mathscr{P}_2(2k+1-i+j), \tag{6.3.9}$$

Using (6.3.8), we see that if $i > 1$ then the resulting term will yield

$$\mu = \overline{(k+3+i)\cdot\mu'} \succ \mu_0 = (k+4, k, k), \tag{6.3.10}$$

and hence, we may ignore these terms.

If $j > 0$, $i = 1$ in (6.3.9), then $\mu' \in \mathscr{P}_2(2k+j)$. The resulting

$$\mu = \overline{(k+4)\cdot\mu'} \succ \mu_0 = (k+4, k, k), \tag{6.3.11}$$

since $j \geq 1$, and hence, we may ignore these terms.

Therefore, the only relevant term in $B^{(3)}$ is for $i = 1$, $j = 0$ and $\mu' = \mu_1^{(2)}(2k) = (k, k)$, in (6.3.9), which yields the term $X(-\mu)$ for

$$\mu = (k+4, k, k) = \mu_6^{(3)}(n). \tag{6.3.12}$$

Therefore, keeping only the relevant terms according to Proposition 4.4.17, we have, modulo $\bar{\mathcal{U}}^{(2)}$,

$$X(-k-3)B^{(3)} = 6X\left(-\mu_6^{(3)}(n)\right) + \dots \tag{6.3.13}$$

since the coefficient to $E'(-1)X(-(k,k))$ in $B^{(2)}(-2k-1)$ is $-1$ (by Lemma 5.4.1).

We now analyze the terms of $X(-k-3)C^{(2)}(-2k-1)$ which are relevant for the computation in Proposition 4.4.17. A typical term in $C^{(2)}(-2k-1)$ is of the form

$$X(-\mu')E'(j), \quad j > 0, \quad \mu' \in \mathscr{P}_2(2k+1+j). \tag{6.3.14}$$

Therefore, the result of the multiplication is

$$X(-k-3)X(-\mu')E'(j), \quad j > 0, \quad \mu' \in \mathscr{P}_2(2k+1+j). \tag{6.3.15}$$

In view of Proposition 4.4.17, we only need to keep track of those terms of the form (6.3.15), such that

$$\mu = \overline{(k+3)\cdot\mu'} \prec \mu_0, \tag{6.3.16}$$

where $\mu' \in \mathscr{P}_2(2k+1+j)$, $j > 0$, and $\mu$ is not reducible by partition larger than $\mu_0 = (k+4,k,k)$.

Recalling the notation from Notation 4.4.13, we have

$$\mathscr{S}_1^{\mu_0} = \left\{ \mu_i^{(3)}(n+1) \,\middle|\, 1 \le i \le 4 \right\} \tag{6.3.17a}$$

$$\mathscr{S}_2^{\mu_0} = \left\{ \mu_i^{(3)}(n+2) \,\middle|\, 1 \le i \le 3 \right\} \tag{6.3.17b}$$

$$\mathscr{S}_3^{\mu_0} = \left\{ \mu_1^{(3)}(n+3), \mu_2^{(3)}(n+3) \right\} \tag{6.3.17c}$$

$$\mathscr{S}_i^{\mu_0} = \left\{ \mu_1^{(3)}(n+i) \right\} \quad \text{for } i = 4,5 \tag{6.3.17d}$$

$$\mathscr{S}_i^{\mu_0} = \emptyset \quad \text{for } i > 5. \tag{6.3.17e}$$

by Proposition 6.2.2 (or from the proof of it), we see that every triplet in $\mathscr{S}_j^{\mu_0}$, for $j \ge 2$, are reducible by partitions larger than $\mu_0$. Thus, we only need the terms involving $\mu \in \mathscr{S}_1^{\mu_0}$.

We have $j = 1$ in the notation of (6.3.16). We list the first few pairs of $2k+2$.

$$\mathscr{P}_2(2k+2) = \{(k+1,k+1),(k+2,k),(k+3,k-1),(k+4,k-2),\dots\}$$

Now if we add $(k+3)$ in these pairs and reorder (if necessary), we get

$$\begin{aligned}(k+3,k+1,k+1) &= \mu_2^{(3)}(n+1),\\ (k+3,k+2,k) &= \mu_3^{(3)}(n+1),\\ (k+3,k+3,k-1) &= \mu_4^{(3)}(n+1),\\ (k+4,k+3,k-2) &\succ \mu_0 \text{ (ignore)}.\end{aligned}$$

We also need to keep track of the coefficients, $-c_b^{(2)}\left(\mu_i^{(2)}(2k+2)\right)$, for $1 \le i \le 3$. From Lemma 5.4.1, we have

$$-c_b^{(2)}\left(\mu_i^{(2)}(2k+2)\right) = \begin{cases} -1 & \text{for } i=1,\\ -2 & \text{for } i>1. \end{cases} \tag{6.3.18}$$

Let us abbreviate,

$$\mu_i = \mu_i^{(3)}(n), \tag{6.3.19a}$$

$$\mu_i' = \mu_i^{(3)}(n+1). \tag{6.3.19b}$$

Therefore, we have (only showing the terms of interest)

$$\begin{aligned}&X(-k-3)C^{(2)}(-2k-1)\\ &\quad = -X(-\mu_2')E'(1) - 2X(-\mu_3')E'(1) - 2X(-\mu_4')E'(1) + \dots.\end{aligned} \tag{6.3.20}$$

We summarize the result of the above computations below.

**Proposition 6.3.1.** *Let $k \ge 3$, and $n = 3k+4$. Let $\mu_i = \mu_i^{(3)}(n)$, and $\mu_i' = \mu_i^{(3)}(n+1)$. Then we have a fifth relation among the homogeneous operators in $\mathcal{U}^{(3)}$ of degree $-n/6$:*

$$R_5^{(3)}(-n) = X(-k-3)R^{(2)}(-2k-1) = 0, \tag{6.3.21}$$

*which can be expressed as*

$$\begin{aligned}&\Big(4X(-\mu_3) + 4X(-\mu_4) + 4X(-\mu_5) + 6X(-\mu_6) + \dots\Big)\\ &\qquad - \Big(X(-\mu_2') + 2X(-\mu_3') + 2X(-\mu_4') + \dots\Big)E'(1) = 0,\end{aligned} \tag{6.3.22}$$

*only showing the terms that are relevant according to Proposition 4.4.17.*

*Remark.* The coefficients in the above relation $R_5^{(3)}(-n)$ are invariant under increasing $k$ by 1 (since the coefficients for degree 2 relations are 1-periodic). The only place, so far, we used the fact $k$ is odd, is to show that $\mathscr{S}_1^{\mu_0}$ contains a partitions that is not reducible (the fourth triplet of $n+1$).

Now, we are ready to use this relation along with the other four regular relations of degree 3.

**Theorem 6.3.2.** *The triplets of the form* $(k+4,k,k)$ *are forbidden for* $k$ *odd. Equivalently, partitions containing difference condition* $[4-,0]$ *are forbidden.*

*Proof.* If $k=1$, $(5,1,1)$ is the fourth triplet of 7, and is forbidden by Proposition 6.2.2.

Assume that $k \geq 3$ odd and set $n = 3k+4$. Let $\mu_* \in \mathscr{P}$ be arbitrary. We set $\mu_0 = \mu_6^{(3)}(n) = (k+4,k,k)$ and $\widetilde{\mu} = \overline{\mu_0\mu_*}$ in the notation of Notation 4.4.3. We need to show that $X(-\widetilde{\mu})v_0 \in V_{(\widetilde{\mu})}$.

For abbreviation, we will use $\mu_j = \mu_j^{(3)}(n)$, and $\mu'_j = \mu_j^{(3)}(n+1)$. Also we will use the following abbreviations for the coefficient functions, $c_i = c_i^{(3)}$ and $c'_i = c_{ib}^{(3)}$, for $i \leq 4$. (Note that $c'_i = 0$ for $i \leq 2$).

Using the five degree 3 relations, we have

$$R_i^{(3)}(-n)X(-\mu_*)v_0 = 0, \tag{6.3.23}$$

for $1 \leq i \leq 5$. We will express the above relations modulo $V_{(\widetilde{\mu})}$ using Proposition 4.4.17. Based, on the discussion in this section, using Proposition 4.4.17, (6.3.23) can be expressed as

$$\sum_{j=1}^{6} c_i(\mu_j)X\left(\overline{\mu_j\mu_*}\right)v_0 - \sum_{j=1}^{4} c'_i(\mu'_j)X(-\mu'_j)E'(1)X(-\mu_*)v_0 \equiv 0 \mod V_{(\widetilde{\mu})}, \tag{6.3.24a}$$

for $1 \leq i \leq 4$, and

$$\sum_{j=1}^{6} a_jX\left(\overline{\mu_j\mu_*}\right)v_0 - \sum_{j=1}^{4} b_jX(-\mu'_j)E'(1)X(-\mu_*)v_0 \equiv 0 \mod V_{(\widetilde{\mu})}, \tag{6.3.24b}$$

where $a_1 = a_2 = 0$, $a_3 = a_4 = a_5 = 4$, $a_6 = 6$, $b_1 = 0$, $b_2 = 1$ and $b_3 = b_4 = 2$, are the constants coming from $R_5^{(3)}(-n)$ as shown in Proposition 6.3.1.

We can further simplify the last four terms in the above equations (6.3.24), using the four regular degree 3 relations $R_i^{(3)}(-n-1)=0$, $i\le 4$, to reduce $X(-\mu'_j)$, if possible. Therefore, we add the relations

$$R_i^{(3)}(-n-1)E'(1)X(-\mu_*)v_0=0, \tag{6.3.25}$$

for $1\le i\le 4$. When expressed the above relations modulo $V_{(\widetilde{\mu})}$, using Proposition 4.4.17, we get

$$\sum_{j=1}^{4} c_i(\mu'_j)X(-\mu'_j)E'(1)X(-\mu_*)v_0\equiv 0 \mod V_{(\widetilde{\mu})}, \tag{6.3.26}$$

for $1\le i\le 4$.

We collect the coefficients of the linear equations in (6.3.24) and (6.3.26) modulo $V_{(\widetilde{\mu})}$ in a $9\times 10$ matrix:

$$\left(\begin{array}{cccccc|cccc}
c_1(\mu_1) & & \dots & \dots & & c_1(\mu_6) & -c'_1(\mu'_1) & \dots & \dots & -c'_1(\mu'_4) \\
\vdots & & \ddots & \ddots & & \vdots & \vdots & \ddots & \ddots & \vdots \\
c_4(\mu_1) & & \dots & \dots & & c_4(\mu_6) & -c'_4(\mu'_1) & \dots & \dots & -c'_4(\mu'_4) \\
0 & 0 & 4 & 4 & 4 & 6 & 0 & -1 & -2 & -2 \\
\hline
0 & & \dots & \dots & & 0 & c_1(\mu'_1) & \dots & \dots & c_1(\mu'_4) \\
\vdots & & \ddots & \ddots & & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & & \dots & \dots & & 0 & c_4(\mu'_1) & \dots & \dots & c_4(\mu'_4)
\end{array}\right). \tag{6.3.27}$$

Notice that if we increase $k$ by 6, the coefficients in the corresponding matrix are exactly the same (by Corollary 5.4.3 and Corollary 5.4.6).

Let $M'(k)$ be the reduced row-echelon form of the above matrix. Then $M'(k)$ is, in fact, invariant under $k\mapsto k+2$ (see § A.1). We present below the matrix $M'(k)$ upto the fifth row:

$$\left(\begin{array}{cccccc|cccc}
1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & -1 \\
0 & 1 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & 3/2 \\
0 & 0 & 1 & 0 & -1/2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 3/2 & 0 & 0 & 0 & 0 & -1/4 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}\right). \tag{6.3.28}$$

Notice that in the fifth row, the coefficients to all terms with positive Heisenberg elements (the last four columns) are zero. Therefore, by Proposition 4.4.17, we have the desired result

$$X(-\widetilde{\mu})v_0 \equiv 0 \mod V_{(\widetilde{\mu})}. \qquad \square$$

## 6.4 Exceptional Forbidden Partitions of Arbitrary Length

In this section, we will prove that partitions of the form

$$\mu_k^m = \Big(k+6+2m,\ \underbrace{k+3+2m,\ \ldots,\ k+3}_{m+1 \text{ parts}},\ k, k\Big) \in \mathscr{P}_{m+4}\left(k(m+4)+(m+3)^2\right), \tag{6.4.1}$$

for $k$ odd and $m \geq 0$ are forbidden, i.e., any partition containing a sub-partition with the difference condition (see Definition 4.1.9) $[3-, 2^*, 3, 0]$, where $2^*$ denotes zero or more occurrence of 2 (for $\mu_k^m$ above 2 is repeated $m$ times in the corresponding difference condition), is reducible. Notice that if $k$ is even then these partition contains the (regular) forbidden triplet $(k+3, k, k) = \mu^{(4)}(3k+3)$ (difference condition $[3-, 0]$), and therefore, $\mu_k^m$ is anyway forbidden if $k$ is even.

The proof of the result for $\mu_k^0$ follows from the degree 2 identity and the results about the exceptional forbidden triplets. For $m > 0$, the result follows from the degree 2 identity and the result for $\mu_{k+2}^{m-1}$. Therefore, it is only natural to prove this result by induction.

Also note that all identities we are using are coming from the degree 2 identity and the degree 3 identities of § 5.3. All coefficients used in the calculation are invariant under increasing every part by 2. (The proofs of Proposition 6.2.1, Proposition 6.2.2 and Theorem 6.3.2 show that, in the row-reduced form, we get equivalent relations if we increase each part by 2). Therefore, it follows that if the result is true for $\mu_k^m$ then it is true for $\mu_{k+2}^m$ as well.

Therefore, it is enough to prove the result for $k = 3$. We choose $k = 3$ instead of $k = 1$ because this is the most general case. If we take $k = 1$, some of the relevant partitions belong to $\mathscr{Q} \setminus \mathscr{P}$, and therefore can be ignored. Also we will see in Chapter 7 that any partition ending with $(1, 1)$ is reducible in all level 4 standard modules because of the initial conditions. Therefore, $\mu_1^m$ will be reducible in any case (notice that $\mu_1^m$ must be the suffix, as it ends with 1).

Let $\pi_k^m$ be the prefix of $\mu_k^m$ of length $m+1$ obtained by omitting the last three parts of $\mu_k^m$,

$$\pi_k^m = \Big(k+6+2m,\ \underbrace{k+3+2m,\ldots,k+5}_{m \text{ parts}}\Big) \in \mathscr{P}_{m+1}\Big(k(m+1)+(m+3)^2-3\Big). \tag{6.4.2}$$

Let $\mu_0' = (k+3,k,k)$, so that $\mu_k^m = \pi_k^m \mu_0'$. Let

$$n = k(m+4)+(m+3)^2, \tag{6.4.3a}$$

$$n' = n-(3k+3) = k(m+1)+(m+3)^2-3, \tag{6.4.3b}$$

$$s = m+4, \tag{6.4.3c}$$

$$s' = s-3 = m+1. \tag{6.4.3d}$$

Then, $\mu_k^m \in \mathscr{P}_s(n)$ and $\pi_k^m \in \mathscr{P}_{s'}(n')$.

Recall the notation $E'(n) = E(-\alpha;n)$. Also recall $R^{(2)}(z)$ from (6.1.1), $A^{(2)}$, $B^{(2)}$ and $C^{(2)}$ from (6.1.7), such that

$$R^{(2)}(-i) \equiv A^{(2)}(-i)+B^{(2)}(-i)+C^{(2)}(-i) \mod \bar{\mathcal{U}}^{(1)}. \tag{6.4.4}$$

for all $i \in \mathbb{N}$. By Proposition 5.3.1, we have

$$R^{(2)}(-i) = 0 \in \operatorname{End} V. \tag{6.4.5}$$

We will now create two relations $R_{m,k}(-n)=0$ and $R'_{m,k}(-n)=0$, where,

$$R_{m,k}(-n) = X(-\pi_k^m)X(-2k-2)R^{(2)}(-2k-1) \in \bar{\mathcal{U}}^{(s)} \tag{6.4.6}$$

$$R'_{m,k}(-n) = X(-\pi_k^m)R^{(2)}(-2k-3)X(-k) \in \bar{\mathcal{U}}^{(s)} \tag{6.4.7}$$

are homogeneous operators of degree $-n/6$.

Recall the tools and techniques described in §4.4. We set $\mu_0 = \mu_k^m$ as in Notation 4.4.3. We will now write out the terms of $R_{m,k}(-n)$ and $R'_{m,k}(-n)$ that are relevant in the context of Proposition 4.4.17, when applying them on $X(-\mu_*)v_0$, for any $\mu_* \in \mathscr{P}$ arbitrary, i.e., the terms in $R_{m,k}(-n)$ and $R'_{m,k}(-n)$ of the form $X(-\mu)$ such that $\mu \preceq \mu_0$, or of the form $X(-\mu)E(j)$ such that $\mu \prec \mu_0$.

We will write $X_{-\mu}$ for $X(-\mu)$, for better readability. Then, using the coefficients from Lemma 5.4.1 and Proposition 2.2.11, we have

$$\begin{aligned}
R_{m,k}(-n) &= X(-\pi_k^m)X_{-(k+2)}R^{(2)}(-2k-1)\\
&= X(-\pi_k^m)X_{-(k+2)}\Big\{\Big[4X_{-(k+1,k)} + 4X_{-(k+2,k-1)} + \dots\Big]\\
&\qquad - E'(-1)\Big[X_{-(k,k)} + \dots\Big]\\
&\qquad - \Big[X_{-(k+1,k+1)} + 2X_{-(k+2,k)} + \dots\Big]E'(1)\\
&\qquad - \Big[2X_{-(k+2,k+1)} + \dots\Big]E'(2)\Big\}\\
&= X(-\pi_k^m)\Big[4X_{-(k+2,k+1,k)} + 4X_{-(k+2,k+2,k-1)} + 6X_{-(k+3,k,k)} + \dots\Big]\\
&\qquad - X(-\pi_k^m)\Big[X_{-(k+2,k+1,k+1)} + X_{-(k+2,k+2,k)}\Big]\\
&= 4X(-\pi_k^m)X_{-(k+2,k+1,k)} + 4X(-\pi_k^m)X_{-(k+2,k+2,k-1)}\\
&\qquad + 6X(-\pi_k^m)X_{-(k+3,k,k)} + \dots
\end{aligned} \tag{6.4.8}$$

Notice, that the terms with positive Heisenberg elements are reducible by partitions larger than $\mu_0$, since the partitions $(k+2,k+1,k)$ and $(k+2,k+2,k-1)$ are reducible by partitions larger than $(k+3,k,k)$ (see the proof of Proposition 6.2.1).

Similar calculation shows that

$$\begin{aligned}
R'_{m,k}(-n) &= X(-\pi_k^m)R^{(2)}(-2k-3)X(-k)\\
&= 4X(-\mu_k^m)X_{-(k+2,k+1,k)} + 6X(-\mu_k^m)X_{-(k+2,k+2,k-1)}\\
&\qquad + 4X(-\mu_k^m)X_{-(k+3,k,k)} + \dots
\end{aligned} \tag{6.4.9}$$

with no significant positive Heisenberg elements (in view of Proposition 4.4.17).

**Proposition 6.4.1** (Base case: $m = 0$)**.** *The partition $\mu_k^0 = (k+6, k+3, k, k)$, for $k > 0$ odd are forbidden.*

*Alternatively, partitions having a sub-partition satisfying the difference condition $[3-, 3, 0]$ are reducible.*

*Proof.* As explained before, it is enough to prove for the case $k = 3$. Note that the coefficients in (6.4.8) and (6.4.9) are the same for all $k$. Therefore, the result is true under the translation $\mu \mapsto \mu + \mathbf{2}$. (we will be using $k$ is odd in the later part of the proof).

Let $R = R(-21) = R_{0,3}(-21)$ and $R' = R'(-21) = R'_{0,3}(-21)$ (see (6.4.6), (6.4.7)). Let $\mu_0 = \mu_3^0$, and $\mu_* \in \mathscr{P}$ be arbitrary (recall Notation 4.4.3 in § 4.4). We will apply Proposition 4.4.17. Note that there are no significant terms with positive Heisenberg elements in either $R(-21)$ or $R'(-21)$. We have

$$\begin{array}{ccccccccccc} R & = & 4X_{-(9,5,4,3)} & + & 4X_{-(9,5,5,2)} & + & 6X_{-(9,6,3,3)} & + & \cdots & = & 0 \\ R' & = & 4X_{-(9,5,4,3)} & + & 6X_{-(9,5,5,2)} & + & 4X_{-(9,6,3,3)} & + & \cdots & = & 0 \end{array} \tag{6.4.10}$$

Subtracting the above equations, we get

$$R'' = R - R' = -2X_{-(9,5,5,2)} + 2X_{-(9,6,3,3)} + \cdots = 0. \tag{6.4.11}$$

Notice that $(9, 5, 5)$ is an exceptional forbidden triplet. Therefore it is reducible by partitions larger than or equal to $(9, 6, 4)$. Therefore, $X_{-(9,5,5,2)}$ is reducible by partition larger than $\mu_0$. (Following the proof of Theorem 6.3.2, we see that this reduction doesn't involve any significant term with positive Heisenberg element). Now, applying Proposition 4.4.17, we get the desired result. (Notice, that it is the reduction of $(9, 5, 5)$ where we need the fact that $k$ is odd.) □

**Theorem 6.4.2.** *Recall $\mu_k^m$ from* (6.4.1)*. The partitions $\mu_k^m$ are forbidden for $m \geq 0$ and $k \geq 1$ odd.*

*Alternatively, any partition having a sub-partition satisfying the difference condition* $[3-, 2^*, 3, 0]$ *(where $2^*$ denotes zero or more occurrence of* 2*) are reducible.*

*Proof.* We prove by induction on $m$. As induction hypothesis we assume that $\mu_{k'}^{m'}$ can be reduced by partition larger than itself without adding any significant term with positive Heisenberg elements (in the context of Proposition 4.4.17), for all $m' < m$ and $k'$ odd.

The base case, $m = 0$, follows from Proposition 6.4.1.

Assume that $m > 0$. We follow the computations done in the proof of Proposition 6.4.1, upto (6.4.11), with $R = R_{m,k}(-n)$, $R' = R'_{m,k}(-n)$, $\mu_0 = \mu_k^m$. Therefore, we have

$$R'' = -2X(-\pi_k^m)X_{-(k+2,k+2,k-1)} + 2X(-\pi_k^m)X_{-(k+3,k,k)} + \cdots = 0 \tag{6.4.12}$$

Notice that

$$\pi_k^m \cdot (k+2, k+2, k-1) = \mu_{k+2}^{m-1} \cdot (k-1), \tag{6.4.13}$$

and

$$\pi_k^m \cdot (k+3, k, k) = \mu_k^m. \tag{6.4.14}$$

Therefore, we can rewrite (6.4.12) as

$$-2X\left(-\mu_{k+2}^{m-1}\right)X(-k+1) + 2X\left(-\mu_k^m\right) + \cdots = 0. \tag{6.4.15}$$

We can reduce the first term (since $k+2$ is odd) in the above equation using induction hypothesis without adding any significant term with positive Heisenberg elements. Notice that if $\mu > \mu_{k+2}^{m-1}$ such that $l(\mu) = l\left(\mu_{k+2}^{m-1}\right)$ and $|\mu| = \left|\mu_{k+2}^{m-1}\right|$ then

$$\mu \succ \mu_{k+2}^{m-1} \succ \pi_k^m, \tag{6.4.16}$$

and therefore,

$$\overline{\mu_{k+2}^{m-1} \cdot (k-1)} > \mu_k^m. \tag{6.4.17}$$

Thus, $\mu_{k+2}^{m-1} \cdot (k-1)$ is reducible by partition larger than $\mu_0 = \mu_k^m$

The result follows from Proposition 4.4.17 (applied on an arbitrary $\mu_* \in \mathscr{P}$). $\square$

## 6.5 Summary of Forbidden Partitions for Level 4 Modules

In this section we summarize the results of this chapter and record some general observations regarding forbidden partitions. Throughout this section, let $V = L(\Lambda)$ be a level 4 standard module with highest weight $\Lambda$, and a highest weight vector $v_0$.

**Notation 6.5.1.** Let $\mathscr{R}^{(\mathrm{L4})} \subset \mathscr{P}$ denote the set of all partitions $\mu \in \mathscr{P}$ such that $\mu$ contains a sub-partition from the following list:

(a) $(4, 1, 1)$,

(b) any partition satisfying one of the following difference conditions:

 (i) $[1]$;

(ii) $[0,0]$, $[0,2]$, $[2,0]$, $[0,3]$, $[3-,0]$, $[0-,4]$, $[4-,0]$;

(iii) $[3-,2^*,3,0]$, where $2^*$ denotes zero or more occurrence of 2.

Let $\mathscr{P}^{(\mathrm{L4})} = \mathscr{P} \setminus \mathscr{R}^{(\mathrm{L4})}$.

*Remark.* $\mathscr{R}^{(\mathrm{L4})}$ is a set of partitions that are reducible for all level 4 standard modules.

We summarize the results of this chapter along with a few useful observations in the following proposition. Recall Notation 4.3.7, Notation 3.2.4.

**Proposition 6.5.2.** (1) *The set*

$$S' = \left\{ \alpha(-\lambda)X(-\mu)v_0 \,\middle|\, \lambda \in \mathscr{O}, \mu \in \mathscr{P}^{(L4)} \right\} \tag{6.5.1}$$

*is a spanning set for any level 4 standard module $V$ with a highest weight vector $v_0$.*

(2) $\mathscr{A}^{\Lambda} \subset \mathscr{P}^{(L4)}$ *and* $F_{\Lambda}(n) \leq \left|\mathscr{P}^{(L4)}(n)\right|$ *for all* $n \geq 0$.

(3) *If $\mu \in \mathscr{P}$ does not contain the sub-partition $(4,1,1)$, then we have*

$$\mu \in \mathscr{R}^{(L4)} \implies \mu + \mathbf{2} \in \mathscr{R}^{(L4)}, \tag{6.5.2}$$

*and for any $\mu \in \mathscr{P}$,*

$$\mu \in \mathscr{P}^{(L4)} \implies \mu + \mathbf{2} \in \mathscr{P}^{(L4)}. \tag{6.5.3}$$

*Proof.* The first two assertions follow from Corollary 4.3.17 in conjunction with Proposition 6.1.1, Proposition 6.2.2, Theorem 6.3.2 and Theorem 6.4.2.

For the third assertion, observe that all the difference conditions in Notation 6.5.1 are invariant under $\mu \mapsto \mu + \mathbf{2}$. Only exception, is the forbidden partition $(4,1,1)$ that does not satisfy any of the difference condition listed above. $\square$

# Chapter 7

# Initial Conditions

In this chapter we will discuss the "initial conditions" for each of the three level 4 standard modules of $\widetilde{\mathfrak{g}}$. These conditions states that a partition is reducible if it contains certain forbidden suffices. Therefore, these conditions do not have the "translation" properties (Proposition 6.5.2(3)) that we saw in the case of forbidden partitions of Chapter 6.

In § 7.1, § 7.2 and § 7.3, we investigate the initial conditions for the $(4,0)$-, $(2,1)$- and the $(0,2)$-module respectively.

Recall that $\omega = e^{i\pi/3}$ and $\omega_0 = e^{i\pi/6}$ are primitive 6-th and 12-th roots of unity respectively. Also recall Definition 4.3.6, Notation 6.5.1.

See Appendix B to find the details of the computer assisted computations used in the following sections.

## 7.1 Initial Condition for the $(4,0)$-module

In this section, let $\Lambda = 4h_0^*$. Let $V = L(\Lambda)$ be the standard module of highest weight $\Lambda$ with a highest weight vector $v_0$. We will show that vectors of the form $\alpha(-\lambda)X(-\mu)v_0$, where $\lambda \in \mathscr{O}$, $\mu \in \mathscr{P}$ having 1 as a part, can be removed from the spanning set (6.5.1).

**Proposition 7.1.1.** *If $\mu \in \mathscr{P}$ is a partition containing 1 as a part, then $\mu$ is reducible.*

*Proof.* Note that if 1 is a part of $\mu$, then it must occur at the end. Therefore, by Proposition 4.4.1, it is enough to show that $X(-1)v_0 \in V_{((1))}$.

On $V$, we have $f_1 v_0 = 0$. Replacing $f_1$ in terms of the vertex operators via Proposition 2.3.1, we get

$$-\frac{2}{\omega_0}X(-1)v_0 + \frac{1}{\sqrt{3}}\alpha(-1)v_0 = 0, \tag{7.1.1}$$

or

$$X(-1)v_0 = \frac{\omega_0\sqrt{3}}{6}\alpha(-1)v_0. \tag{7.1.2}$$

Thus, the result follows. □

*Remark.* Recall Notation 6.5.1. We could also prove this by comparing our spanning set (6.5.1) against the graded dimension using Corollary 4.3.17. We have $F_\Lambda(1) = 0$, but $|\mathscr{P}^{(\mathrm{L4})}(1)| = 1$. ($\mathscr{P}^{(\mathrm{L4})}(1) = \{(1)\}$). Therefore $X(-1)v_0$ must be in $V_{((1))} = V^{(0)}$.

## 7.2 Initial Conditions for the $(2,1)$-Module

In this section, let $\Lambda = 2h_0^* + h_1^*$. Let $V = L(\Lambda)$ be the standard module of highest weight $\Lambda$ with a highest weight vector $v_0$. In this section we will show that vectors of the form $\alpha(-\lambda)X(-\mu)v_0$, where $\lambda \in \mathscr{O}$, $\mu \in \mathscr{P}$ having 1, 2 or 3 twice as a part, can be removed from the spanning set (6.5.1).

We used Maple programs to straighten out various monomials in the proofs below. These programs and the Maple worksheet used to carry out these computations are presented in Appendix B.

**Lemma 7.2.1.** *In the* $(2,1)$*-module* $V$*, we have* $X(-1)^2v_0 \in V_{((1,1))}$.

*Proof.* In the $(2,1)$-module, we have $f_1^2 v_0 = 0$. We can write this relation in terms of the $\alpha(\bullet)$ and $X(\bullet)$ operators using Proposition 2.3.1, and then straighten out the terms using the bracket formulae in Proposition 4.2.2. This yields the following relation

$$X(-1)^2v_0 = -\frac{\omega_0\sqrt{3}}{6}X(-2)v_0 + \frac{\omega_0}{\sqrt{3}}\alpha(-1)X(-1)v_0 - \frac{\omega}{12}\alpha(-1)^2v_0. \tag{7.2.1}$$

Since the RHS belongs to $V_{((1,1))}$, this gives us the desired result. □

*Remark.* Alternatively, we could have used Corollary 4.3.17 to argue that we have an extra partition in $\mathscr{P}^{(\mathrm{L4})}(2)$ (recall Notation 6.5.1). Therefore, one of the partitions, (2) or $(1,1)$ must be reducible in the $(2,1)$-module. If we assume that (2) is reducible, then by Proposition 4.4.1, any partition ending with a 2 must also be reducible. Let $\mathscr{P}'$ be the set of partitions in $\mathscr{P}^{(\mathrm{L4})}$ not containing any partition ending with a 2. We get

$2 = |\mathscr{P}'(7)| < F_{(2,1)}(7) = 3$, contradicting Corollary 4.3.17. Therefore, (2) cannot be reducible, and by elimination, $(1,1)$ must be reducible.

**Lemma 7.2.2.** *In the* $(2,1)$*-module* $V$*, we have* $X(-2)^2 v_0 \in V_{((2,2))}$.

*Proof.* In the $(2,1)$-module, we have $f_0^3 v_0 = 0$. We rewrite this relation in terms of the $\alpha(\bullet)$ and $X(\bullet)$ operators using Proposition 2.3.1. Then we apply the relation (7.2.1) and straighten out the terms using the bracket relations in Proposition 4.2.2. We obtain

$$X(-2)X(-1)v_0 = -\frac{\omega_0}{\sqrt{3}}\alpha(-1)X(-2)v_0 + \frac{\omega_0\sqrt{3}}{2}\alpha(-1)^2X(-1)v_0. \tag{7.2.2}$$

This does not give us anything new—it just shows that $(2,1)$ is a reducible partition. However, we are going to use this to simplify the next relation, $f_0^4 v_0 = 0$. Once again we rewrite this relation in terms of the $\alpha(\bullet)$ and $X(\bullet)$ operators, apply the rewriting rules (7.2.1) and (7.2.2), and straighten out the terms using the bracket relations in Proposition 4.2.2. This gives us the following relation:

$$\begin{aligned} X(-2)^2v_0 = &-\frac{4}{3}X(-3)X(-1)v_0 - \frac{\omega_0\sqrt{3}}{18}X(-4)v_0 \\ &+ \frac{\omega_0 2\sqrt{3}}{9}\alpha(-1)X(-3)v_0 - \omega_0\sqrt{3}\alpha(-1)^2X(-2)v_0 \\ &+ \frac{\omega_0 4}{\sqrt{3}}\alpha(-1)^3X(-1)v_0 + \frac{\omega}{12}\alpha(-1)^4v_0. \end{aligned} \tag{7.2.3}$$

Thus, the desired result follows. □

*Remark.* Alternatively, we could use Corollary 4.3.17 and Proposition 4.4.1 to prove that $(2,2)$ is reducible. We have an extra partition in $\mathscr{P}^{(\mathrm{L4})}(4)$ after removing reducible partitions of 4 ending with $(1,1)$. The partitions in $\mathscr{P}^{(\mathrm{L4})}(4)$ that does not end with $(1,1)$ are $(2,2)$, $(3,1)$ and $(4)$. If we assume that $(4)$ is reducible, then we get a contradiction for the partitions of $n = 8$. If we assume that $(3,1)$ is reducible then we get a contradiction for the partitions of $n = 9$. Therefore, by elimination, $(2,2)$ must be reducible.

**Lemma 7.2.3.** *In the* $(2,1)$*-module* $V$*,* $X(-3)^2v_0 \in V_{((3,3))}$.

*Proof.* The proof is similar to that of Lemma 7.2.2. We will use the following two additional relations in the $(2,1)$-module:

$$f_0^5 v_0 = 0, \qquad f_0^6 v_0 = 0.$$

From $f_0^5 v_0 = 0$, expressing it in terms of the operators $\alpha(\bullet)$ and $X(\bullet)$, simplifying and straightening out the terms using the bracket relations and the rewriting rules (7.2.1), (7.2.2), (7.2.3), we get

$$\begin{aligned} X(-3)X(-2)v_0 = {} & -X(-4)X(-1)v_0 + \alpha(-1)X(-3)X(-1)v_0 \\ & - \frac{\omega_0\sqrt{3}}{6}\alpha(-1)X(-4)v_0 + \frac{\omega_0}{\sqrt{3}}\alpha(-1)^2X(-3)v_0 \\ & - \frac{\omega_0\sqrt{3}}{2}\alpha(-1)^3X(-2)v_0 + \frac{\omega_0\sqrt{3}}{2}\alpha(-1)^4X(-1)v_0 \\ & + \frac{\omega}{20}\alpha(-1)^5v_0 + \frac{\omega}{30}\alpha(-5)v_0. \end{aligned} \tag{7.2.4}$$

From $f_0^6 v_0 = 0$, expressing it in terms of the operators $\alpha(\bullet)$ and $X(\bullet)$, simplifying and straightening out the terms using the bracket relations and the rewriting rules (7.2.1), (7.2.2), (7.2.3), (7.2.4), we get

$$\begin{aligned} X(-3)^2v_0 = {} & 2X(-5)X(-1)v_0 - \frac{\omega_0\sqrt{3}}{6}X(-6)v_0 \\ & - 6\alpha(-1)X(-4)X(-1)v_0 + \frac{\omega_0 2}{\sqrt{3}}\alpha(-1)X(-5)v_0 \\ & + 6\alpha(-1)^2X(-3)X(-1)v_0 - \omega_0\sqrt{3}\alpha(-1)^2X(-4)v_0 \\ & + \omega_0\sqrt{3}\alpha(-1)^3X(-3)v_0 - \frac{\omega_0 3\sqrt{3}}{5}\alpha(-1)^5X(-1)v_0 \\ & - \frac{\omega_0\sqrt{3}}{15}\alpha(-5)X(-1)v_0 - \frac{\omega}{6}\alpha(-5)\alpha(-1)v_0. \end{aligned} \tag{7.2.5}$$

Since the terms on the RHS of the above equation belong to $V_{((3,3))}$, the result follows. □

*Remark.* We were unable to find a proof by contradiction and elimination (as we could for the previous two cases) based on Proposition 4.4.1 and Corollary 4.3.17. We could not find any contradiction if we assumed that (6) is reducible. This is why we decided to give a direct proof.

**Proposition 7.2.4.** *In the* $(2,1)$*-module any partition ending with* $(1,1)$*,* $(2,2)$ *and* $(3,3)$ *are reducible.*

*Proof.* By Proposition 4.4.1, it is enough to show that $(1,1)$, $(2,2)$ and $(3,3)$ are reducible. Therefore, the result follows from Lemma 7.2.1, Lemma 7.2.2 and Lemma 7.2.3. □

## 7.3 Initial Conditions for the $(0,2)$-Module

In this section, let $\Lambda = h_0^* + 2h_1^*$. Let $V = L(\Lambda)$ be the standard module of highest weight $\Lambda$ with a highest weight vector $v_0$. We will show that partitions having 1 or 3 as a part, or having 2 as a part twice are reducible. Furthermore, any partition ending with $(5,2)$, $(7,4,2)$, $(9,6,4,2)$, $\ldots$ etc. are also reducible.

**Lemma 7.3.1.** *In the* $(0,2)$*-module* $V$*, we have* $X(-1)v_0 \in V^{(0)}$.

*Proof.* In the $(0,2)$-module, we have $f_0 v_0 = 0$. Using Proposition 2.3.1, we have

$$X(-1)v_0 = -\frac{\omega_0}{\sqrt{3}}\alpha(-1)v_0. \tag{7.3.1}$$

□

**Lemma 7.3.2.** *In the* $(0,2)$*-module* $V$*, we have* $X(-3)v_0 \in V_{((3))}$.

*Proof.* In the $(0,2)$-module we have $f_1^3 v_0 = 0$. Expressing this relation in terms of the operators $\alpha(\bullet)$ and $X(\bullet)$, simplifying and straightening out using the bracket relations Proposition 4.2.2 and the rewriting rule (7.3.1), we get

$$X(-3)v_0 = \frac{3}{2}\alpha(-1)X(-2)v_0 + \frac{\omega_0\sqrt{3}}{4}\alpha(-1)^3 v_0. \tag{7.3.2}$$

Thus, the result follows. □

*Remark.* The above two lemmas could also be argued based on the graded dimension formula, using Proposition 4.4.1 and Corollary 4.3.17.

**Lemma 7.3.3.** *In the* $(0,2)$*-module* $V$*, we have* $X(-2)^2 v_0 \in V_{((2,2))}$.

*Proof.* In the $(0,2)$-module V, we have $f_1^4 v_0 = 0$. Expressing this relation in terms of the operators $\alpha(\bullet)$ and $X(\bullet)$, simplifying and straightening out using the bracket relations Proposition 4.2.2 and the rewriting rules (7.3.1) and (7.3.2), we get

$$X(-2)^2 v_0 = -\frac{\omega_0\sqrt{3}}{2}X(-4)v_0 + \omega_0\sqrt{3}\alpha(-1)^2 X(-2)v_0 + \frac{\omega 3}{4}\alpha(-1)^4 v_0. \tag{7.3.3}$$

Therefore, the result follows immediately. □

*Remark.* The above lemma could also be argued based on the graded dimension formula, using Proposition 4.4.1 and Corollary 4.3.17. We have an extra partition in $\mathscr{P}^{(\mathrm{L4})}(4)$ after removing partitions ending with a 1 or a 3. Therefore, one of the two partitions $(2,2)$ or $(4)$ must be reducible. If we assume that $(4)$ is reducible, then we arrive at a contradiction for the partitions of $n = 8$.

The following initial conditions are consequence of the interplay of Lemma 7.3.2 and other reducible partitions in $\mathscr{R}^{(\mathrm{L4})}$ that ends with a 3.

**Lemma 7.3.4.** *In the* $(0,2)$*-module* $V$*, the following partitions*

$$(5,2),(7,4,2),(9,6,4,2),(11,8,6,4,2),\ldots, etc..$$

*are reducible. Notice that all these partitions satisfy the difference condition* $[3,2^*]$ *(here* $2^*$ *denotes zero or more occurrence of* $2$*), and end with the lowest part* $2$*.*

*Proof.* First, we will show that $X(-5)X(-2)v_0 \in V_{((5,2))}$. Using the degree 2 relation $R^{(2)}(-7)v_0 = 0$ (6.1.3), we have

$$X(-4)X(-3)v_0 + X(-5)X(-2)v_0 \equiv 0 \mod V_{((5,2))}. \tag{7.3.4}$$

Applying the operator $X(-4)$ on both sides of (7.3.2), we also have

$$X(-4)X(-3)v_0 \equiv \frac{3}{2}X(-5)X(-2)v_0 \mod V_{((5,2))}. \tag{7.3.5}$$

Combining (7.3.4) and (7.3.5) gives us the desired result.

Now, we will prove the result for $(7,4,2)$. We proceed in the same fashion as in the proof of Theorem 6.3.2, except this time we need to keep track of terms involving partitions upto $(7,4,2)$ (one additional term). See the corresponding matrix in row-reduced form in § A.1. The result shows that,

$$X(-(7,3,3))v_0 \equiv 0 \mod V_{((7,4,2))}. \tag{7.3.6}$$

However, applying $X(-7)X(-3)$ on both sides of (7.3.2), we get

$$X(-(7,3,3))v_0 \equiv \frac{3}{2}X(-(7,4,2))v_0 \mod V_{((7,4,2))}. \tag{7.3.7}$$

Combining (7.3.7) and (7.3.6) gives us the desired result.

Now, we will prove the general case. Let $\mu_0$ be a partition satisfying the difference condition $[3,2^*]$ (with at least two occurrence of 2) and ending with a 2.

We will follow similar computations as done in §6.4. Recall the notations $\mu_k^m$ (6.4.1), $\pi_k^m$ (6.4.2), $R_{m,k}$ (6.4.6) and $R'_{m,k}$ (6.4.7). Also recall (6.4.3).

Let $\mu_2^m$ be the partition obtained by replacing the suffix $(4,2)$ of $\mu_0$ by $(3,3)$, where $m = l(\mu_0) - 4 \geq 0$. Then, we have $l\left(\mu_2^m\right) = l(\mu_0)$, and $|\mu_2^m| = |\mu_0|$. And, $\mu_0$ is the next smallest partition of $n$ into $s$ parts after $\mu_2^m$.

Therefore, we need to keep track of an extra term in each of (6.4.8) and (6.4.9). Therefore, we have (applying to $v_0$)

$$\begin{aligned}4X(-\pi_2^m)X_{-(5,4,3)}v_0 &+ 4X(-\pi_2^m)X_{-(5,5,2)}v_0 \\ &+ 6X(-\pi_2^m)X_{-(6,3,3)}v_0 + 12X(-\pi_2^m)X_{-(6,4,2)}v_0 \equiv 0 \mod V_{(\mu_0)},\end{aligned} \tag{7.3.8}$$

$$\begin{aligned}4X(-\pi_2^m)X_{-(5,4,3)}v_0 &+ 6X(-\pi_2^m)X_{-(5,5,2)}v_0 \\ &+ 4X(-\pi_2^m)X_{-(6,3,3)}v_0 + 12X(-\pi_2^m)X_{-(6,4,2)}v_0 \equiv 0 \mod V_{(\mu_0)}.\end{aligned} \tag{7.3.9}$$

Subtracting (7.3.9) from (7.3.8), we get

$$-2X(-\pi_2^m)X_{-(5,5,2)}v_0 + 2X(-\pi_2^m)X_{-(6,3,3)}v_0 \equiv 0 \mod V_{(\mu_0)}. \tag{7.3.10}$$

Notice that $\pi_k^m \cdot (5,5) = \mu_{k+2}^{m-1}$, and therefore, the first term in the above equation can be reduced by partitions larger than $\mu_0$. Also, $\pi_k^m \cdot (6,3,3) = \mu_k^m$. Following the same argument, as in the proof of Theorem 6.4.2, we obtain

$$X(-\pi_2^m)X_{-(6,3,3)}v_0 \in V_{(\mu_0)}. \tag{7.3.11}$$

However, by (7.3.2), we have

$$X(-\pi_2^m)X_{-(6,3,3)}v_0 \equiv \frac{3}{2}X(-\pi_2^m)X_{-(6,4,2)}v_0 \mod V_{(\mu_0)}. \tag{7.3.12}$$

But $\pi_k^m \cdot (6,4,2) = \mu_0$. Therefore, combining (7.3.11) and (7.3.12), we obtain the desired result:

$$X(-\mu_0)v_0 \in V_{(\mu_0)}. \qquad \square$$

**Proposition 7.3.5.** *For the* $(0,2)$*-module, any partition ending with a* $1$ *or* $3$*, or ending with the sub-partition* $(2,2)$ *or one of the following sub-partitions:*

$$(5,2),(7,4,2),(9,6,4,2),(11,8,6,4,2),\ldots,etc..,$$

*(i.e., sub-partitions satisfying the difference condition* $[3,2^*]$ *and ending with a* $2$*) is reducible.*

*Proof.* The result follows from Lemma 7.3.1, Lemma 7.3.2, Lemma 7.3.3, Lemma 7.3.4 and Proposition 4.4.1. $\square$

# Chapter 8

# Partition Identities

In this chapter, we summarize our main results and propose three new partition identities. These results prove one inequality of the the proposed identities. We have also verified the partition identities for partitions of $n$, for $n \leq 170$, and $n = 180, 190$ and 200. The C program used for the verification is included in Appendix C. This demonstrates a strong evidence for the validity of these partition identities.

## 8.1 The Main Result

Let $V = L(\Lambda)$ be a level 4 standard module for $\widetilde{\mathfrak{g}}$ of highest weight $\Lambda$, and a highest weight vector $v_0$, where $\Lambda = (4,0), (2,1)$ or $(0,2)$. Recall Definition 4.1.9, Notation 3.2.4, Notation 4.3.7 and Notation 6.5.1.

Let $\mathscr{R}'^{(4,0)} \subset \mathscr{P}$ be the set of all partitions in $\mathscr{R}^{(\mathrm{L4})}$ and all partitions that end with $(1)$. Let $\mathscr{P}^{(4,0)} = \mathscr{P} \setminus \mathscr{R}'^{(4,0)}$.

Let $\mathscr{R}'^{(2,1)} \subset \mathscr{P}$ be the set of all partitions in $\mathscr{R}^{(\mathrm{L4})}$ and all partitions that end with $(1,1)$, $(2,2)$ or $(3,3)$. Let $\mathscr{P}^{(2,1)} = \mathscr{P} \setminus \mathscr{R}'^{(2,1)}$.

Let $\mathscr{R}'^{(0,2)} \subset \mathscr{P}$ be the set of all partitions in $\mathscr{R}^{(\mathrm{L4})}$ and all partitions that end with $(1)$, $(3)$, $(2,2)$, or that end with a sub-partition ending with a 2 and satisfying the difference condition $[3, 2^*]$, where $2^*$ denotes zero or more occurrence of 2. Let $\mathscr{P}^{(0,2)} = \mathscr{P} \setminus \mathscr{R}'^{(0,2)}$.

**Theorem 8.1.1.** *Let $V = L(\Lambda)$ be a level 4 standard module for $\widetilde{\mathfrak{g}}$ of highest weight $\Lambda$, and a highest weight vector $v_0$, where $\Lambda = (4,0), (2,1)$ or $(0,2)$. Then, with the notations described above, the set*

$$S_\Lambda = \left\{ \alpha(-\lambda)X(-\mu)v_0 \,\middle|\, \lambda \in \mathscr{O},\, \mu \in \mathscr{P}^\Lambda \right\} \tag{8.1.1}$$

*is a spanning set for $V$. Furthermore,*

$$F_\Lambda(n) \leq \left|\mathscr{P}^\Lambda(n)\right|, \tag{8.1.2}$$

*for all $n \geq 0$. The equality holds in* (8.1.2) *for all $n \geq 0$ if and only if the set* (8.1.1) *is a basis for $V$. (The equality is verified for $n \leq 170$ and $n = 180, 190$ and 200.)*

*Proof.* From Proposition 6.5.2, Proposition 7.1.1, Proposition 7.2.4 and Proposition 7.3.5, we have

$$\mathscr{R}'^\Lambda \subset \mathscr{R}^\Lambda,$$

and therefore,

$$\mathscr{A}^\Lambda \subset \mathscr{P}^\Lambda.$$

Thus, the result follows from Corollary 4.3.17. ☐

Paraphrasing the above theorem, the three (conjectured) partition identities are presented below.

**$\boldsymbol{\Lambda} = (\mathbf{4}, \mathbf{0})$:** The number of partitions of $n \geq 0$ with parts congruent to $\pm 2, \pm 3, \pm 4$ modulo 14 is less than or equal to the number of partitions of $n$ into parts greater than 1, *and* having no sub-partition with difference condition $[1]$, $[0,0]$, $[0,2]$, $[2,0]$, $[0,3]$, $[3-,0]$, $[0-,4]$, $[4-,0]$ or $[3,2^*,3,0]$. The equality holds for all $n \geq 0$ if and only if the set (8.1.1) is a basis of $V = L(4,0)$. Furthermore, the equality has been verified to hold for $n \leq 170$, $n = 180, 190$ and 200.

**$\boldsymbol{\Lambda} = (\mathbf{2}, \mathbf{1})$:** The number of partitions of $n \geq 0$ with parts congruent to $\pm 1, \pm 4, \pm 6$ modulo 14 is less than or equal to the number of partitions of $n$ not ending with $(1,1)$, $(2,2)$ or $(3,3)$, *and* having no sub-partition with difference condition $[1]$, $[0,0]$, $[0,2]$, $[2,0]$, $[0,3]$, $[3-,0]$, $[0-,4]$, $[4-,0]$ or $[3,2^*,3,0]$. The equality holds for all $n \geq 0$ if and only if the set (8.1.1) is a basis of $V = L(2,1)$. Furthermore, the equality has been verified to hold for $n \leq 170$, $n = 180, 190$ and 200.

**$\boldsymbol{\Lambda} = (\mathbf{0}, \mathbf{2})$:** The number of partitions of $n \geq 0$ with parts congruent to $\pm 2, \pm 5, \pm 6$ modulo 14 is less than or equal to the number of partitions of $n$ not ending with $(1)$, $(2,2)$ or $(3)$, *and* not ending with a partition satisfying the difference condition $[3,2^*]$ that ends with $(2)$, *and* having no sub-partition with difference condition $[1]$, $[0,0]$, $[0,2]$,

$[2,0]$, $[0,3]$, $[3-,0]$, $[0-,4]$, $[4-,0]$ or $[3,2^*,3,0]$. The equality holds for all $n \geq 0$ if and only if the set (8.1.1) is a basis of $V = L(0,2)$. Furthermore, the equality has been verified to hold for $n \leq 170$, $n = 180, 190$ and $200$.

# Appendix A

# Computation of the Relations

In this appendix, we present the maple programs that we used to compute the relations and their computations.

In § A.1, we present the Maple worksheet showing examples of our calculations that were used in Chapter 6 and Chapter 7. The worksheet uses the codes from three other Maple source files presented in the subsequent sections.

In § A.2, we present the Maple source file containing the programs used to generate the list of partitions of a positive integer $n$ into $k$ parts in the decreasing lexicographical order (see Notation 4.1.17). The algorithm we implemented is from [Cha11].

In § A.3, we present the Maple source file containing the programs used to compute the coefficients of $X(-\mu)$ in various product generating functions, as described in § 5.4.

In § A.4, we present the Maple source file containing various procedures used to automate our calculations of the relations used in the Maple worksheet presented in § A.1.

## A.1 Maple Worksheet for Computing Relations

In this section, we present the Maple worksheet showing examples of our calculations that were used in Chapter 6 and Chapter 7. The worksheet uses the codes from three other Maple source files presented in the subsequent sections. All three source files (named, `chat.txt`, `coeffs.txt` and `test.txt`) must be saved in the same working directory as this Maple worksheet.

```
> read `cha.txt`:
  read `coef.txt`:
  read `test.txt`:
> with(LinearAlgebra):

> # Examples of degree 3 relations.
> for n from 3 to 16 do
    print(n);
    print(truncate(listkPartitions(n,3),4));
    print(deg3(n));
    print("----------------------");
  end do:
```

3

[[1, 1, 1]]

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

"----------------------"

4

[[2, 1, 1]]

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

"----------------------"

5

[[2, 2, 1], [3, 1, 1]]

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

"----------------------"

6

[[2, 2, 2], [3, 2, 1], [4, 1, 1]]

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

"-----------------------"

7

[[3, 2, 2], [3, 3, 1], [4, 2, 1], [5, 1, 1]]

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

"-----------------------"

8

[[3, 3, 2], [4, 2, 2], [4, 3, 1], [5, 2, 1]]

$$\begin{bmatrix} 1 & 0 & 0 & -4 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

"-----------------------"

9

[[3, 3, 3], [4, 3, 2], [4, 4, 1], [5, 2, 2]]

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

"-----------------------"

10

[[4, 3, 3], [4, 4, 2], [5, 3, 2], [5, 4, 1]]

$$\begin{bmatrix} 1 & 0 & 0 & -4 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

"-----------------------"

11

[[4, 4, 3], [5, 3, 3], [5, 4, 2], [5, 5, 1]]

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

"-----------------------"

12

[[4, 4, 4], [5, 4, 3], [5, 5, 2], [6, 3, 3]]

$$\begin{bmatrix} 1 & 0 & 0 & -9 \\ 0 & 1 & 0 & \frac{5}{2} \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

"-----------------------"

13

[[5, 4, 4], [5, 5, 3], [6, 4, 3], [6, 5, 2]]

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

"-----------------------"

14

[[5, 5, 4], [6, 4, 4], [6, 5, 3], [6, 6, 2]]

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & \frac{3}{2} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

"-----------------------"

15

[[5, 5, 5], [6, 5, 4], [6, 6, 3], [7, 4, 4]]

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

"-----------------------"

16

$$[[6,5,5],[6,6,4],[7,5,4],[7,6,3]]$$

$$\begin{bmatrix} 1 & 0 & 0 & -4 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

"-----------------------" **(1)**

```
> # Computation of the Matrix from the proof of the degree 3
  # exceptional triplets.
  #
  for n from 13 to 25 by 6 do
    print(n);
    print(truncate(listkPartitions(n,3),6));
    print(truncate(listkPartitions(n+1,3),4));
    print(deg3ex(n));
    print("-----------------------");
  end do:
```

13

$$[[5,4,4],[5,5,3],[6,4,3],[6,5,2],[6,6,1],[7,3,3]]$$

$$[[5,5,4],[6,4,4],[6,5,3],[6,6,2]]$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & \frac{3}{2} \\ 0 & 0 & 1 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{3}{2} & 0 & 0 & 0 & 0 & -\frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{3}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

"-----------------------"

19

$$[[7,6,6],[7,7,5],[8,6,5],[8,7,4],[8,8,3],[9,5,5]]$$

$$[[7,7,6],[8,6,6],[8,7,5],[8,8,4]]$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & \frac{3}{2} \\ 0 & 0 & 1 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{3}{2} & 0 & 0 & 0 & 0 & -\frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{3}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

"-----------------------"

25

[[9, 8, 8], [9, 9, 7], [10, 8, 7], [10, 9, 6], [10, 10, 5], [11, 7, 7]]

[[9, 9, 8], [10, 8, 8], [10, 9, 7], [10, 10, 6]]

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & \frac{3}{2} \\ 0 & 0 & 1 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{3}{2} & 0 & 0 & 0 & 0 & -\frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{3}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

"-----------------------" **(2)**

```
> # Computation of the matrix for the initial condition
  # of the (0,2)-module showing that (7,4,2) is reducible.
  #
  n := 13;
  L := truncate(listkPartitions(n, 3), 7);
  for i from 1 to 4 do
    r[i] := map(c||i, L);
  end do:
```

```
r[5] := [0$7]:
L1 := listkPartitions(7,2);
for i from 1 to 7 do
  for x in L1 do
    if L[i] = sort([6,op(x)], `>`) then
      r[5][i] := r[5][i] + c(x);
    end if;
  end do:
end do:
L2 := listkPartitions(6,2);
for i from 1 to 7 do
  for x in L2 do
    if L[i] = sort([7,op(x)], `>`) then
      r[5][i] := r[5][i] + 6*cb(x);
    end if;
  end do:
end do:
M := Matrix([seq(r[i], i=1..5)]):
ReducedRowEchelonForm(M);
```

$$n := 13$$

$$L := [[5,4,4],[5,5,3],[6,4,3],[6,5,2],[6,6,1],[7,3,3],[7,4,2]]$$

$$L1 := [[4,3],[5,2],[6,1]]$$

$$L2 := [[3,3],[4,2],[5,1]]$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 2 & 0 & 2 \\ 0 & 1 & 0 & 0 & -3 & 0 & -6 \\ 0 & 0 & 1 & 0 & -\frac{1}{2} & 0 & 2 \\ 0 & 0 & 0 & 1 & \frac{3}{2} & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (3)$$

## A.2 Maple Codes to Generate Partitions

In this section, we present the source file containing the codes used to generate the list of partitions of a positive integer $n$ into $k$ parts in the decreasing lexicographical order (see Notation 4.1.17). The algorithm we implemented is from [Cha11]. Partitions are represented as a non-increasing list of positive integers.

Listing A.1: cha.txt

```
#################################################################
#
# File: cha.txt
#
# Author: Debajyoti Nandi
#
# Generating Partitions of n into k parts, based on
#   Sung-Huyk Cha, "Recursive algorithms for generating
#   partitions of an integer", 2011
#
# Link:
#  http://support.csis.pace.edu/CSISWeb/docs/techReports/
#  techReport280.pdf
#
#
# The main functions:
#   listkPartitions(n,k): lists the partitions of n into k
#     parts in ascending lexicographic order. A partition is
#     represented as a non-increasing list (aka, descending
#     composition).
#
#   Example: listkPartitions(8,3)
#   => [[3,3,2],[4,2,2],[4,3,1],[5,2,1],[6,1,1]]
#
#   allPartitions(n): lists all partitions of n, in ascending
#     order (with respect to ">" on P). Each partition is
#     represented as a non-increasing list.
#
#   Example: allPartitions(5)
#   => [[1,1,1,1,1],[2,1,1,1],[2,2,1],[3,1,1],[3,2],[4,1],[5]]
#
#################################################################
```

```
# The maximum number to be partitioned
_N := 100;

# The list to hold the current partition being generated
_p := [0$_N];

# The list to hold the generated partitions
_L := [];

# resetL(): Resets the global variable _L
_resetL := proc()
  global _L := [];
end proc;

# _P(n,k,visit): - Generates partitions of n into k parts,
#   and calls  the function visit() each time a partition is
#   generated.
#
_P := proc(n,k,visit)
  if n >= k then
    _R(n, k, n-k+1, 1, visit);
  end if;
end proc;

# _R(n,k,s,t,visit): Recursive backbone of _P(), generates
#   partitions of n into k parts, with the largest part s,
#   (t is the position index where this partitions is to be
#   added in _p). visit() is as above.
#
# The following invariant is always true:
#   ceil(n/k) <= s <= n-k+1.
#
_R := proc(n,k,s,t,visit)
  global _p;
  local i;

  if k=1 then
    _p[t] := n;
    visit(t);
    return;
```

```
  end if;

  for i from ceil(n/k) to s do
    _p[t] := i;
    _R(n-i, k-1, min(i, n-i-k+2), t+1, visit);
  end do;
end proc;

# _listIt(t) - The visitor function that puts the constructed
#   partition into the list _L, t is the length of the current
#   partition.
#
_listIt := proc(t)
  global _p, _L;
  _L := [op(_L), _p[1..t]];
end proc;

# listkPartitions(n,k): Returns a list of partitions of n into
#   k parts. The partitions are represented as non-increasing
#   lists. The partitions are arranged in ascending
#   lexicographical order.
#
# Example: listkPartitions(8,3)
# => [[3,3,2],[4,2,2],[4,3,1],[5,2,1],[6,1,1]]
#
listkPartitions := proc(n,k)
  global _L;
  local L;
  _restetL();
  _P(n,k,_listIt);
  L := _L;
  _resetL();
  return L;
end proc;

#   allPartitions(n): lists all partitions of n, in ascending
#     order (with respect to ">" on P). Each partition is
#     represented as a non-increasing list.
#
#   Example: allPartitions(5)
#   => [[1,1,1,1,1],[2,1,1,1],[2,2,1],[3,1,1],[3,2],[4,1],[5]]
```

```
#
allPartitions := proc(n)
  local L := [], k;
  for k from n to 1 by -1 do
    L := [op(L), op(listkPartitions(n,k))];
  end do;
  return L;
end proc;
```

## A.3 Maple Codes to Compute the Coefficients

Here, we present the source file containing the codes used to compute the coefficients of $X(-\mu)$ in various product generating functions, as described in § 5.4.

Listing A.2: coeffs.txt

```
################################################################
#
# File: coeffs.txt
#
# Author: Debajyoti Nandi
#
# This file includes code for generating the coefficients
# of X(-L) in various generating function identities, L is
# a partition of n.
#
# _genCoeff(deg, Spec, L):
#  This procedure computes the coefficient of X(-L) in the
#  product generating function (of degree deg)
#     X(v^i.a)X(v^j.a)...
#  deg = # of factors
#  Spec = [i,j...] the powers of nu (v) that appears above
#  L = a partition of n into deg parts.
#
# ca(L): coefficient of X(-L) (L is a partition into two parts)
#        in X(a,a) [Spec=[0,0]];
#
# cb(L): coefficient of X(-L) (L is a partition into two parts)
#        in X(-a,-a) [Spec=[3,3]]
#
# c(L): coefficient of X(-L) (L is a partition into two parts)
```

```
#        in X(a,a) - E^-(-a)X(-a,-a)E^+(a)
#
# c1a(L): coefficient of X(-L) (L is a triplet) in
#         X(a,a,a) [Spec=[0,0,0]]
#
# c1b(L): 0 (the LHS has deg = 1 < 3)
#
# c1(L): same as c1a
#
# c2a(L): coefficient of X(-L) (L is a triplet) in
#         X(a,a,v.a) [Spec=[0,0,1]]
#
# c2b(L): 0 (the LHS has deg = 2 < 3)
#
# c2(L): same as c2a;
#
# c3a(L): coefficient of X(-L) (L is a triplet) in
#         X(a,v.a,v.a) [Spec=[0,1,1]]
#
# c3b(L): coefficient of X(-L) (L is a triplet) in
#         X(-a,v^2.a,v^2.a) [Spec=[3,2,2]]
#
# c3(L): coefficient of X(-L) (L is a triplet) in
#         X(a,a,v.a) - E^-(-a)X(-a,v^2.a,v^2.a)E^+(-a)
#
# c4a(L): coefficient of X(-L) (L is a triplet) in
#         X(a,v.a,v^(-1).a) [Spec=[0,1,-1]]
#
# c4b(L): coefficient of X(-L) (L is a triplet) in
#         X(-a,v^2.a,v^(-2).a) [Spec=[3,2,-2]]
#
# c4(L): coefficient of X(-L) (L is a triplet) in
#        X(a,v.a,v^(-1).a) - E^-(-a)X(-a,v^2.a,v^(-2).a)E^+(-a)
#
###############################################################

# _genCoeff(deg, Spec, L):
#  This procedure computes the coefficient of X(-L) in the
#  product generating function (of degree deg)
#     X(v^i.a)X(v^j.a)...
#  deg = # of factors
```

```
#  Spec = [i,j...] the powers of nu (v) that appears above
#  L = a partition of n into deg parts.
#
_genCoeff := proc(deg, Spec, L)
  local i, x, m, M, L1, Vars, C, P;
  local w := exp(2*Pi*I/6);

  if deg <> nops(Spec) or deg <> nops(L) then
    return FAIL;
  end if;

  # distinct elements of L
  L1 := [op({op(L)})];

  # m[x] = multiplicity of x in L
  for x in L1 do
    m[x] := nops(select(`=`, L, x));
  end do;

  # sort the list of multiplicities
  M := sort([seq(m[x], x in L1)],`>`);

  # sort the parts in L by multiplicities.
  L1 := sort(L1, (x,y)-> m[x]>m[y] or (m[x]=m[y] and x>y));

  # generating the appropriate polynomial
  Vars := [seq(`x`||i, i=1..nops(L1))];
  for i from 1 to deg do
    C[i] := [seq(w^(-Spec[i]*x), x in L1)];
    P[i] := `+`(seq(C[i][j]*Vars[j], j=1..nops(L1)));
  end do;
  P[0] := `*`(seq(P[i], i=1..deg));

  # return the appropriate coefficient
  return evalc(coeftayl(P[0], Vars = [0$nops(L1)], M));
end proc;

################################################################
#
# Coefficients in the degree 2 identities:
#
```

```
ca := L -> _genCoeff(2, [0,0], L);
cb := L -> _genCoeff(2, [3,3], L);
c := L -> evalc(ca(L)-cb(L));

##################################################################
#
# Coefficients in the degree 3 identities:
#

c1a := L -> _genCoeff(3, [0,0,0], L);
c1b := L -> 0;
c1 := c1a;

c2a := L -> _genCoeff(3, [0,0,1], L);
c2b := L -> 0;
c2 := c2a;

c3a := L -> _genCoeff(3, [0,1,1], L);
c3b := L -> _genCoeff(3, [3,2,2], L);
c3 := L -> evalc(c3a(L) - c3b(L));

c4a := L -> _genCoeff(3, [0,1,-1], L);
c4b := L -> _genCoeff(3, [3,2,-2], L);
c4 := L -> evalc(c4a(L) - c4b(L));
```

## A.4 Other Codes Used in the Maple Worksheet

We present below the Maple source file containing various procedures used to automate our calculations of the relations used in the Maple worksheet presented in § A.1.

Listing A.3: test.txt

```
##################################################################
#
# File: test.txt
#
# Author: Debajyoti Nandi
#
# The programs here can be used to automate the analysis
# of various deg 2 and 3 relations. Used in the proofs of
```

```
# various forbidden partitions.
#
# truncate(L,k): returns a truncated list from L upto k
#    elements
#
# deg3(n): Uses the 4 relations of degree 3 on the least 4
#    partitions of n. Returns a row reducede matrix
#    from the coefficients. j-th column corresponds
#    to the the j-th least partition of n into 3 parts.
#
# deg3ex(n): Computes the row reduced matrix in the
#    calculation of the exceptional triplets. Input should
#    be of the form n = 3k+1, n >= 13. The j-th column,
#    1 <= j <= 6, corresponds to the term X(-Lj),
#    where Lj is the j-th least partition of n into 3 parts.
#    The j-th column, 7 <= j <= 10, corresponds to the term
#    X(-Lj)E(1), where Lj is the j-th least partition of
#    (n+1) into k parts.
#
################################################################

# truncate(L,k): truncates the list L up to length k
truncate := (L,k) -> if k<nops(L) then L[1..k] else L end if;

# deg3(n): presents the row-reduced form of the deg-3 relations
#    upto the term corresponding to the 4th triplet of n.
#
deg3 := proc(n)
  local L, i, r;
  L := truncate(listkPartitions(n,3),4);
  for i from 1 to 4 do
    r[i] := map(c||i, L);
  end do;
  return ReducedRowEchelonForm(Matrix([seq(r[i], i=1..4)]));
end proc;

# deg3ex(n): computes the row-reduced matrix used in the
#    proof of the exceptional forbidden triplets of n.
#
deg3ex := proc(n)
  local L1, L2, S1, S2, S3, k, i, x, r;
```

```
# We must have: n = 3*k+4, n >= 13
if n mod 3 <> 1 or n < 13 then
  return FAIL;
end if;

k := (n-4)/3;

# first 4 rows
L1 := truncate(listkPartitions(n,3), 6);
L2 := truncate(listkPartitions(n+1,3), 4);
for i from 1 to 4 do
  r[i] := [op(map(c||i, L1)), op(map(-c||i||b, L2))];
end do;

# 5th row
S1 := listkPartitions(2*k+1, 2);
r[5] := [0$10];
for i from 1 to 6 do
  for x in S1 do
    if L1[i] = sort([k+3, op(x)],`>`) then
      r[5][i] := r[5][i] + c(x);
    end if;
  end do;
end do;
S2 := listkPartitions(2*k, 2);
for i from 1 to 6 do
  for x in S2 do
    if L1[i] = sort([k+4, op(x)],`>`) then
      r[5][i] := r[5][i] - (-6)*cb(x);
    end if;
  end do;
end do;
S3 := listkPartitions(2*k+2, 2);
for i from 1 to 4 do
  for x in S3 do
    if L2[i] = sort([k+3, op(x)],`>`) then
      r[5][6+i] := r[5][6+i] - cb(x);
    end if;
  end do;
end do;
```

```
  # 6th-9th rows
  for i from 6 to 9 do
    r[i] := [0$6, op(map(c||(i-5), L2))];
  end do;

  #matrix
  return ReducedRowEchelonForm(Matrix([seq(r[i], i=1..9)]));
end proc;
```

# Appendix B

# Computation in Noncommutative Algebra

In this appendix, we present the maple programs we used to straighten out monomials in non-commuting variables. We have used the data structure and algorithms in NCFPS (noncommutative formal power series) package [Zei12] of D. Zeilberger with minor modifications (also see [BRRZ12]). The algorithm to apply substitution rules to straighten out an out-of-order monomial is based on the algorithm and Maple codes of M. Russell (see [Rus13]). His program was for finitely many substitution rules over a finite alphabet. We modified Russel's code to implement infinitely many rules (based on finitely many patterns) over an infinite indexed alphabet.

In § B.1, we present the Maple worksheet to verify the isomorphism in Proposition 2.3.1. In § B.2, we present the Maple worksheet to carry out the computations used in the proofs of various initial conditions in Chapter 7.

The above Maple worksheets require other Maple source files for manipulating formal polynomials in non-commuting variables and applying substitution rules. We also need the Maple source files implementing all the substitution rules that we require for our computations.

In § B.3, we present the Maple source files to manipulate and straighten out formal polynomials in non-commuting indexed variables. In § B.4, we present our Maple source files implementing the substitution rules that we require for our computations. In § B.5, we present our Maple source files containing miscellaneous useful procedures used in the above mentioned Maple worksheets. All these supporting Maple source files must be saved in the same working directory as the Maple worksheets.

## B.1 Verification of the Isomorphism

In this section, we present the worksheet to verify the isomorphism in Proposition 2.3.1. The worksheet requires the files `npolyio.txt`, `npolyops.txt`, `npolysubs.txt` from § B.3; the file `A22-rules.txt` from § B.4; and the file `misc.txt` from § B.5. These files should be saved in the same directory as the worksheet.

```
> read `npolyio.txt`:
  read `npolyops.txt`:
  read `npolysubs.txt`:
  read `A22-rules.txt`:
  read `misc.txt`:
```

```
> w := exp(Pi*I/3):
  w0 := exp(Pi*I/6):
```

```
> h0 := parsePoly(h[0]): h1 := parsePoly(h[1]):
  e0 := parsePoly(e[0]): e1 := parsePoly(e[1]):
  f0 := parsePoly(f[0]): f1 := parsePoly(f[1]):
```

```
> H0 := evalc(rewritePoly(h0,Risom)):
  H1 := evalc(rewritePoly(h1,Risom)):
  E0 := evalc(rewritePoly(e0,Risom)):
  E1 := evalc(rewritePoly(e1,Risom)):
  F0 := evalc(rewritePoly(f0,Risom)):
  F1 := evalc(rewritePoly(f1,Risom)):
```

### Checking: $[h_i, h_j] = 0, \quad (0 \le i, j \le 1)$.

```
> seq(seq(
    writePoly(evalcPoly(rewritePoly(b(H||i, H||j), Rvop))),
  i=0..1), j=0..1);
```

$$0, 0, 0, 0 \tag{1}$$

### Checking: $[h_i, e_j] - A_{ij} e_j = 0, \quad (0 \le i, j \le 1)$.

```
> seq(seq(
    writePoly(evalcPoly(rewritePoly(
      addPoly(b(H||i, E||j), sMulPoly(-A22[i,j],E||j)), Rvop))),
  i=0..1), j=0..1);
```

$$0, 0, 0, 0 \tag{2}$$

### Checking: $[h_i, f_j] + A_{ij} e_j = 0, \quad (0 \le i, j \le 1)$.

```
> seq(seq(
    writePoly(evalcPoly(rewritePoly(
      addPoly(b(H||i, F||j), sMulPoly(A22[i,j],F||j)), Rvop))),
  i=0..1), j=0..1);
```

$$0, 0, 0, 0 \tag{3}$$

### Checking: $[e_i, f_j] - \delta_{ij} h_j = 0, \quad (0 \le i, j \le 1)$.

```
> seq(seq(
    writePoly(evalcPoly(rewritePoly(
      addPoly(b(E||i, F||j),
```

```
        sMulPoly(-delta(i,j),H||i)), Rvop))),
  i=0..1), j=0..1);
```

$$0, 0, 0, 0 \quad (4)$$

### Checking: $\left(ad\ e_i\right)^{-A_{ij}+1} e_j = 0, \quad (i \neq j)$.

```
> Y := E;
  i,j := 0,1:
  writePoly(evalcPoly(rewritePoly(adpow(-A22[i,j]+1, X||i, X||j),
  Rvop)));
  i,j := 1,0:
  writePoly(evalcPoly(rewritePoly(adpow(-A22[i,j]+1, X||i, X||j),
  Rvop)));
```

$$Y := E$$

0

0 **(5)**

### Checking: $\left(ad\ f_i\right)^{-A_{ij}+1} f_j = 0, \quad (i \neq j)$.

```
> Y := F;
  i,j := 0,1:
  writePoly(evalcPoly(rewritePoly(adpow(-A22[i,j]+1, X||i, X||j),
  Rvop)));
  i,j := 1,0:
  writePoly(evalcPoly(rewritePoly(adpow(-A22[i,j]+1, X||i, X||j),
  Rvop)));
```

$$Y := F$$

0

0 **(6)**

## B.2 Computations for the Proofs of the Initial Conditions

In this section, we present the Maple worksheet to carry out the computations used in the proofs of various initial conditions in Chapter 7. The worksheet requires the files `npolyio.txt`, `npolyops.txt`, `npolysubs.txt` from § B.3; the files `A22-rules.txt` and `A22-L4-iniRules.txt` from § B.4; and the file `misc.txt` from § B.5. These files should be saved in the same directory as the worksheet.

Note that as we discover a new relation in this worksheet, we have added them to the file `A22-L4-iniRules.txt` progressively.

# Setup

```
> read `npolyio.txt`:
  read `npolyops.txt`:
  read `npolysubs.txt`:
  read `A22-rules.txt`:
  read `misc.txt`:
  read `A22-L4-iniRules.txt`:
```

```
> w := exp(Pi*I/3):
  w0 := exp(Pi*I/6):
```

# For the (4,0)-module

# We have, $f_1 v_0 = 0$.

```
> p := f[1].v[0]: P:= parsePoly(p): is0inV(P,"40");
```

$$true \tag{2.1}$$

```
> Q := rewritePoly(P, Risom union Rvop);
```

$$Q := \left[\left[-\frac{4}{\sqrt{3}+\mathrm{I}}, [X_{-1}, v_0]\right], \left[\frac{1}{3}\sqrt{3}, [a_{-1}, v_0]\right]\right] \tag{2.2}$$

```
> # Normalize
  Q0 := sMulPoly((sqrt(3)+I)/4, Q): writePoly(Q0);
```

$$-X_{-1}.v_0 + \frac{1}{12}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\ (a_{-1}.v_0) \tag{2.3}$$

The above relations shows that (1) is reducible.

# For the (2,1)-module

# We have, $f_1^2 v_0 = 0$.

```
> p := (f[1]^2).v[0]: P := parsePoly(p): is0inV(P,"21");
```

$$true \tag{3.1}$$

```
> Q := rewritePoly(P, Risom union Rvop);
```

$$Q := \left[\left[\frac{16}{\left(\sqrt{3}+\mathrm{I}\right)^2}, [X_{-1}, X_{-1}, v_0]\right], \left[-\frac{8}{3}\frac{\sqrt{3}}{\sqrt{3}+\mathrm{I}}, [a_{-1}, X_{-1}, v_0]\right], \left[\frac{1}{3},\right.\right.$$
$$\left.\left.[a_{-1}, a_{-1}, v_0]\right], \left[\frac{4}{3}\frac{\sqrt{3}}{\sqrt{3}+\mathrm{I}}, [X_{-2}, v_0]\right]\right] \tag{3.2}$$

# Normalize

```
> Q0 := sMulPoly(-(sqrt(3)+I)^2/16,Q): writePoly(Q0);
```

$$-\left(X_{-1}^2\right).v_0 + \frac{1}{6}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\ (a_{-1}.X_{-1}.v_0) - \frac{1}{48}\left(\sqrt{3}+\mathrm{I}\right)^2\left(\left(a_{-1}^2\right).v_0\right) \tag{3.3}$$

$$-\frac{1}{12}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\left(X_{-2}.v_0\right)$$

The above relation shows that (1,1) is reducible.
We have added the above rule as 'rini21a' in A22-L4-iniRules.txt.

# We have, $f_0^3 v_0 = 0$.
We will add rini21a in our substitution rules to reduce the above.

```
> p := (f[0]^3).v[0]: P := parsePoly(p): is0inV(P,"21");
```

$$true \tag{3.4}$$

```
> Q := rewritePoly(P, Risom union Rvop union {rini21a});
```

$$Q := \left[\left[\frac{72\sqrt{2}}{\sqrt{3}+\mathrm{I}}, \left[a_{-1}, a_{-1}, X_{-1}, v_0\right]\right], \left[-\frac{96\sqrt{2}\sqrt{3}}{\left(\sqrt{3}+\mathrm{I}\right)^2}, \left[X_{-2}, X_{-1}, v_0\right]\right], \left[-\frac{48\sqrt{2}}{\sqrt{3}+\mathrm{I}}, \left[a_{-1}, X_{-2}, v_0\right]\right]\right] \tag{3.5}$$

```
> # Normalize Q
  Q0 := sMulPoly((sqrt(3)+I)^2/(96*sqrt(6)), Q):
  writePoly(Q0);
```

$$\frac{1}{4}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\left(\left(a_{-1}^2\right).X_{-1}.v_0\right) - X_{-2}.X_{-1}.v_0 - \frac{1}{6}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\left(a_{-1}.X_{-2}.v_0\right) \tag{3.6}$$

We have added the above rule as 'rini21b' in A22-L4-iniRules.txt.

# We have, $f_0^4 v_0 = 0$.
We will add rini21b in our substitution rules to reduce the above.

```
> p := (f[0]^4).v[0]: P := parsePoly(p): is0inV(P,"21");
```

$$true \tag{3.7}$$

```
> Q := rewritePoly(P, Risom union Rvop
          union {rini21a, rini21b});
```

$$Q := \left[\left[-\frac{384\sqrt{3}}{\sqrt{3}+\mathrm{I}}, \left[a_{-1}, a_{-1}, a_{-1}, X_{-1}, v_0\right]\right], \left[-12, \left[a_{-1}, a_{-1}, a_{-1}, a_{-1}, v_0\right]\right], \left[\frac{288\sqrt{3}}{\sqrt{3}+\mathrm{I}}, \left[a_{-1}, a_{-1}, X_{-2}, v_0\right]\right], \left[\frac{768}{\left(\sqrt{3}+\mathrm{I}\right)^2}, \left[X_{-3}, X_{-1}, v_0\right]\right], \left[\frac{576}{\left(\sqrt{3}+\mathrm{I}\right)^2}, \left[X_{-2}, X_{-2}, v_0\right]\right], \left[-\frac{64\sqrt{3}}{\sqrt{3}+\mathrm{I}}, \left[a_{-1}, X_{-3}, v_0\right]\right], \left[-4\,\mathrm{I}\sqrt{3}\left(\mathrm{I}\sqrt{3}+1\right), \left[X_{-4}, v_0\right]\right]\right] \tag{3.8}$$

```
> # Normalize
  Q0 := sMulPoly(-(sqrt(3)+I)^2/576, Q): writePoly(Q0);
```

$$\frac{2}{3}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\left(\left(a_{-1}^{3}\right).X_{-1}.v_0\right)+\frac{1}{48}\left(\sqrt{3}+\mathrm{I}\right)^2\left(\left(a_{-1}^{4}\right).v_0\right)-\frac{1}{2}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\left(\left(a_{-1}^{2}\right).X_{-2}.v_0\right)-\frac{4}{3}X_{-3}.X_{-1}.v_0-\left(X_{-2}^{2}\right).v_0+\frac{1}{9}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\left(a_{-1}.X_{-3}.v_0\right)+\frac{1}{144}\mathrm{I}\left(\sqrt{3}+\mathrm{I}\right)^2\sqrt{3}\left(\mathrm{I}\sqrt{3}+1\right)\left(X_{-4}.v_0\right) \quad (3.9)$$

The above relation shows that (2,2) is reducible.
We have added the above rule as 'rini21c' in A22-L4-iniRules.txt.

# We have, $f_0^{\,5} v_0 = 0$.
We will add rini21c in our substitution rules to reduce the above.

```
> p := (f[0]^5).v[0]: P := parsePoly(p): is0inV(P,"21");
```

$$\mathit{true} \quad (3.10)$$

```
> Q := rewritePoly(P, Risom union Rvop
         union {rini21a, rini21b, rini21c});
```

$$Q := \left[\left[-\frac{1440\sqrt{2}}{\sqrt{3}+\mathrm{I}},\left[a_{-1},a_{-1},a_{-1},a_{-1},X_{-1},v_0\right]\right],\left[-24\sqrt{2}\sqrt{3},\left[a_{-1},a_{-1},a_{-1},a_{-1},a_{-1},v_0\right]\right],\left[\frac{1440\sqrt{2}}{\sqrt{3}+\mathrm{I}},\left[a_{-1},a_{-1},a_{-1},X_{-2},v_0\right]\right],\left[-\frac{1920\sqrt{2}\sqrt{3}}{\left(\sqrt{3}+\mathrm{I}\right)^2},\left[a_{-1},X_{-3},X_{-1},v_0\right]\right],\left[-\frac{960\sqrt{2}}{\sqrt{3}+\mathrm{I}},\left[a_{-1},a_{-1},X_{-3},v_0\right]\right],\left[\frac{1920\sqrt{2}\sqrt{3}}{\left(\sqrt{3}+\mathrm{I}\right)^2},\left[X_{-4},X_{-1},v_0\right]\right],\left[\frac{1920\sqrt{2}\sqrt{3}}{\left(\sqrt{3}+\mathrm{I}\right)^2},\left[X_{-3},X_{-2},v_0\right]\right],\left[\frac{480\sqrt{2}}{\sqrt{3}+\mathrm{I}},\left[a_{-1},X_{-4},v_0\right]\right],\left[-16\sqrt{2}\sqrt{3},\left[a_{-5},v_0\right]\right]\right] \quad (3.11)$$

```
> # Normalize
  Q0 := sMulPoly(-(sqrt(3)+I)^2/(1920*sqrt(6)), Q):
  writePoly(Q0);
```

$$\frac{1}{4}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\left(\left(a_{-1}^{4}\right).X_{-1}.v_0\right)+\frac{1}{80}\left(\sqrt{3}+\mathrm{I}\right)^2\left(\left(a_{-1}^{5}\right).v_0\right)-\frac{1}{4}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\left(\left(a_{-1}^{3}\right).X_{-2}.v_0\right)+a_{-1}.X_{-3}.X_{-1}.v_0+\frac{1}{6}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\left(\left(a_{-1}^{2}\right).X_{-3}.v_0\right)-X_{-4}.X_{-1}.v_0-X_{-3}.X_{-2}.v_0-\frac{1}{12}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\left(a_{-1}.X_{-4}.v_0\right)+\frac{1}{120}\left(\sqrt{3}+\mathrm{I}\right)^2\left(a_{-5}.v_0\right) \quad (3.12)$$

We have added the above rule as 'rini21d' in A22-L4-iniRules.txt.

# We have, $f_0^{\,6} v_0 = 0$.
We will add rini21d in our substitution rules to reduce the above.

```
> p := (f[0]^6).v[0]: P := parsePoly(p): is0inV(P,"21");
```

$$true \tag{3.13}$$

```
> Q := rewritePoly(P, Risom union Rvop
        union {rini21a, rini21b, rini21c});
```

$$Q := \left[\left[-\frac{12672\sqrt{3}}{\sqrt{3}+I}, [a_{-1}, a_{-1}, a_{-1}, a_{-1}, a_{-1}, X_{-1}, v_0]\right], [-576, [a_{-1}, a_{-1}, a_{-1}, a_{-1}, a_{-1}, a_{-1}, v_0]], \left[\frac{11520\sqrt{3}}{\sqrt{3}+I}, [a_{-1}, a_{-1}, a_{-1}, a_{-1}, X_{-2}, v_0]\right], \left[-\frac{23040}{(\sqrt{3}+I)^2}, [a_{-1}, a_{-1}, X_{-3}, X_{-1}, v_0]\right], \left[-\frac{5760\sqrt{3}}{\sqrt{3}+I}, [a_{-1}, a_{-1}, a_{-1}, X_{-3}, v_0]\right], \left[\frac{23040}{(\sqrt{3}+I)^2}, [a_{-1}, X_{-4}, X_{-1}, v_0]\right], \left[\frac{46080}{(\sqrt{3}+I)^2}, [a_{-1}, X_{-3}, X_{-2}, v_0]\right], \left[\frac{1920\sqrt{3}}{\sqrt{3}+I}, [a_{-1}, a_{-1}, X_{-4}, v_0]\right], \left[\frac{7680}{(\sqrt{3}+I)^2}, [X_{-5}, X_{-1}, v_0]\right], \left[-\frac{3840}{(\sqrt{3}+I)^2}, [X_{-3}, X_{-3}, v_0]\right], [32\, I\sqrt{3}\,(I\sqrt{3}+1), [a_{-5}, X_{-1}, v_0]], [-544, [a_{-5}, a_{-1}, v_0]], \left[\frac{1280\sqrt{3}}{\sqrt{3}+I}, [a_{-1}, X_{-5}, v_0]\right], \left[-\frac{320\sqrt{3}}{\sqrt{3}+I}, [X_{-6}, v_0]\right]\right] \tag{3.14}$$

```
> Q0 := sMulPoly((sqrt(3)+I)^2/3840, Q): writePoly(Q0);
```

$$-\frac{33}{10}(\sqrt{3}+I)\sqrt{3}\,((a_{-1}^5).X_{-1}.v_0) - \frac{3}{20}(\sqrt{3}+I)^2((a_{-1}^6).v_0) + 3(\sqrt{3}+I)\sqrt{3}\,((a_{-1}^4).X_{-2}.v_0) - 6\,((a_{-1}^2).X_{-3}.X_{-1}.v_0) - \frac{3}{2}(\sqrt{3}+I)\sqrt{3}\,((a_{-1}^3).X_{-3}.v_0) + 6\,(a_{-1}.X_{-4}.X_{-1}.v_0) + 12\,(a_{-1}.X_{-3}.X_{-2}.v_0) + \frac{1}{2}(\sqrt{3}+I)\sqrt{3}\,((a_{-1}^2).X_{-4}.v_0) + 2\,(X_{-5}.X_{-1}.v_0) - (X_{-3}^2).v_0 + \frac{1}{120}\,I\,(\sqrt{3}+I)^2\sqrt{3}\,(I\sqrt{3}+1)\,(a_{-5}.X_{-1}.v_0) - \frac{17}{120}(\sqrt{3}+I)^2(a_{-5}.a_{-1}.v_0) + \frac{1}{3}(\sqrt{3}+I)\sqrt{3}\,(a_{-1}.X_{-5}.v_0) - \frac{1}{12}(\sqrt{3} \tag{3.15}$$

$+ \mathrm{I}) \sqrt{3} \ (X_{-6}.v_0)$

The above relations shows that (3,3) is reducible.

## For the (0,2)-module

# We have, $f_0^2 v_0 = 0$.

```
> p := f[0].v[0]: P := parsePoly(p): is0inV(P,"02");
```

$$true \tag{4.1}$$

```
> Q := rewritePoly(P, Risom union Rvop);
```

$$Q := \left[\left[\frac{4\sqrt{2}}{\sqrt{3}+\mathrm{I}}, [X_{-1}, v_0]\right], \left[\frac{2}{3}\sqrt{2}\sqrt{3}, [a_{-1}, v_0]\right]\right] \tag{4.2}$$

```
> # Normalize
  Q0 := sMulPoly(-(sqrt(3)+I)/(4*sqrt(2)), Q):
  writePoly(Q0);
```

$$-X_{-1}.v_0 - \frac{1}{6}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\ (a_{-1}.v_0) \tag{4.3}$$

The above relations shows that (1) is reducible.
We have added the above rule as 'rini02a' in A22-L4-iniRules.txt.

# We have, $f_1^3 v_0 = 0$.
We will add rini02a in our substitution rules to reduce the above.

```
> p := (f[1]^3).v[0]: P := parsePoly(p): is0inV(P,"02");
```

$$true \tag{4.4}$$

```
> Q := rewritePoly(P, Risom union Rvop union {rini02a});
```

$$Q := \left[\left[3\sqrt{3}, [a_{-1}, a_{-1}, a_{-1}, v_0]\right], \left[\frac{36}{\sqrt{3}+\mathrm{I}}, [a_{-1}, X_{-2}, v_0]\right], \left[-\frac{24}{\sqrt{3}+\mathrm{I}}, [X_{-3}, v_0]\right]\right] \tag{4.5}$$

```
> # Normalize
  Q0 := sMulPoly((sqrt(3)+I)/24, Q): writePoly(Q0);
```

$$\frac{1}{8}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\ \left((a_{-1}^3).v_0\right) + \frac{3}{2}\ a_{-1}.X_{-2}.v_0 - X_{-3}.v_0 \tag{4.6}$$

The above relations shows that (3) is reducible.
We have added the above rule as 'rini02b' in A22-L4-iniRules.txt.

# We have, $f_1^4 v_0 = 0$.
We will add rini02b in our substitution rules to reduce the above.

```
> p := (f[1]^4).v[0]: P := parsePoly(%): is0inV(P,"02");
```

(4.7)

$$true \tag{4.7}$$

```
> Q := rewritePoly(P, Risom union Rvop
        union {rini02a, rini02b});
```

$$Q := \left[\left[-27, \left[a_{-1}, a_{-1}, a_{-1}, a_{-1}, v_0\right]\right], \left[-\frac{72\sqrt{3}}{\sqrt{3}+\mathrm{I}}, \left[a_{-1}, a_{-1}, X_{-2}, v_0\right]\right], \left[\frac{144}{\left(\sqrt{3}+\mathrm{I}\right)^2}, \left[X_{-2}, X_{-2}, v_0\right]\right], \left[\frac{36\sqrt{3}}{\sqrt{3}+\mathrm{I}}, \left[X_{-4}, v_0\right]\right]\right] \tag{4.8}$$

```
> # Normalize
  Q0 := sMulPoly(-(sqrt(3)+I)^2/120, Q): writePoly(Q0);
```

$$\frac{9}{40}\left(\sqrt{3}+\mathrm{I}\right)^2\left(\left(a_{-1}^4\right).v_0\right)+\frac{3}{5}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\left(\left(a_{-1}^2\right).X_{-2}.v_0\right)-\frac{6}{5}\left(X_{-2}^2\right).v_0 - \frac{3}{10}\left(\sqrt{3}+\mathrm{I}\right)\sqrt{3}\left(X_{-4}.v_0\right) \tag{4.9}$$

The above relaton shows that (2,2) is forbidden.

## B.3 Maple Codes for Noncommutative Polynomials

In § B.3, we present the Maple source files to manipulate and straighten out formal polynomials in non-commuting variables. The data structure and algorithms for various operations on noncommutative polynomials are based on the NCFPS (noncommutative formal power seires) package of D. Zeilberger [Zei12, BRRZ12].

Here we list three Maple files: `npolyio.txt`, `npolyops.txt` and `npolysubs.txt`.

We present our Maple program to convert a noncommutative polynomial between mathematical notation and the internal data structure (using lists, see [Zei12]) below.

Listing B.1: npolyio.txt

```
#################################################################
#
# File: npolyio.txt
#
# Author: Debajyoti Nandi
#
# Input and Output of Noncommutative Polynomials
# ==============================================
#
# This maple programs read a polynomial (assumed noncommutative)
# in certain format and converts it to an internal data structure
# representing this polynomial. The data structure used is the
# same as that used in NCFPS of Zeilberger:
#
# Link: http://www.math.rutgers.edu/~zeilberg/tokhniot/NCFPS
#
# The purpose of this file is to make the input/output
# of noncommutative polynomials easier to human beings.
#
# Data Structure
# ==============
#
# Monomial: A monomial is a (noncommutative) product in
# indeterminates. A monomial is represented as a list.
# For example, x.y.y.x -> [x,y,y,x]
#
# Term: A term is a constant times a monomial, i.e., t=c*m,
#   where c is a constant and m is a monomial. A term is
```

```
#   represented as t = [c, m].
#   For example, -2x.y.x -> [-2, [x,y,x]].
#
# Polynomial: A polynomial is a sum of terms, represented as a
#   list of monomials: p = [t1,t2,...,tn]
#   For example, x.y-y.x -> [[1,[x,y]], [-1,[y,x]]].
#
# Input Format
# ============
#
# Constant terms: A constant term must be inputed as c*Id.
#
# Terms: If the coefficient is 1, then it can be ommited.
# Otherwise, input in the form: (examples)
#     3*(x.y.y.x), (y^2).z, a*(x.x.y.x), 2*Id etc.
#   Note that if a symbol is used for the constant it is assumed
#   to be a commutative symbol ("a" in the third example above)
#   A bracket must be used if more than one symbol is multiplied
#   in the monomial.
#
# Polynomial: (Example)
#   3*(x.y.y.x) - 2*x + z.x.y + 5*Id.
#   3*((x^2).y.z)
#
# This polynomial will be read and converted ot internal data
# structure as:
#   [[3,[x,y,y,x]], [-2,[x]], [1,[z,x,y]], [5,[]]]
#   [[3,[x,x,y,z]]]
#
# Output Format
# =============
#
# An internal representation of a noncommutative polynomial
# is converted back to maple expression using '*' and '.'.
# For example,
#   [[2,[]], [-1,[x,y]] + [1,[y,y]]] --> 2 - x.y + y^2.
#
# Provides
# ========
#
# parsePoly(f): returns the internal representation of the
```

```
#   noncommutative polynomial f (see "input format" above).
#
# writePoly(P): returns an expression using '*' and '.'
#   to express the internal representation of a noncommutative
#   polynomial P in better-to-read format (see "output format").
#
# Disclaimer
# ==========
#
# No sanity check is done. No error message is issued if
# the input is invalid (not conforming to the "input format"
# above).
#
#################################################################

### Input ###

# splitPow(p): If p is a power in an indeterminate, then it splits
#   it into a sequence of factors.
# Example: x^3 -> x,x,x

splitPow := proc(p)
  if p::`^` then
    return seq(op(1,p), i=1..op(2,p));
  else
    return p;
  end if;
end proc;


# splitMono(m): Splits a monomial m, returning a list of its
#   factors.
# Examples: x.x.y.x -> [x,x,y,x], Id -> []

splitMono := proc(m)
  if m = Id then
    return [];
  elif m::`.` then
    return map(splitPow, [op(1.., m)]);
  else
    return [splitPow(m)];
```

```
  end if;
end proc;


# splitTerm(t): Splits a term into a list with the coefficient
#   as the first element, and the monomial as the second.
# Examples:
#   2*(x.y) -> [2, x.y], 3*Id -> [3, Id], x.y.y -> [1, x.y^2]

splitTerm := proc(t)
  if t::`*` then
    return [`*`(op(1..nops(t)-1, t)), op(-1,t)];
  else
    return [1,t];
  end if;
end proc;


# splitPoly(f): Splits a polynomial f into terms
# Example: x.y.y.x + 2*x - Id -> [x.y^2.x, 2*x, -Id]

splitPoly := proc(f)
  if f::`+` then
    return [op(1.., f)];
  else
    return [f];
  end if;
end proc;


# parsePoly(f): reads a polynomial in the input format, and
#   returns its representation in the above data structure.
# Example:
#    x.y.y.x + 2*x - Id -> [[1, [x,y,y,x]], [2, [x]], [-1, []]]

parsePoly := proc(f)
  local P := [], T, t, L;

  T := splitPoly(f);
  for t in T do
    L := splitTerm(t);
```

```
    if t <> 0 then
      P := [op(P), [L[1], splitMono(L[2])]];
    end if;
  end do;

  return P;
end proc;


#################################################################

### Output ###

# writeMono(m): expresses a monomial using `.`
# Example: [x,y,y,x] -> x.y^2.x

writeMono := proc(m)
  return `.`(op(m));
end proc;

# writeTerm(t): expresses a term as prduct of the constant and
#   the monomial using `*`
# Example: [a, [x.y.y.x]] -> a * x.y^2.x

writeTerm := proc(t)
  return t[1] * writeMono(t[2]);
end proc;

# writePoly(f): expresses an internal representation of a
#   polynomial f in human readable format.
# Example:
#   [[1, [x,y,y,x]], [2, [x]], [-1, []]] -> x.y^2.x + 2*x - 1

writePoly := proc(F)
  local L := map(writeTerm, F);
  return `+`(op(L));
end proc;
```

The file below is a slight modification of Zeilberger's NCFPS package [Zei12].

Listing B.2: npolyops.txt

```
#################################################################
```

```
#
# File: npolyops.txt
#
# Operations on Noncommutative Polynomials
# ========================================
#
# Author: Debajyoti Nandi
#
# Inspired by Zeilberger (NCFPS)
#   http://www.math.rutgers.edu/~zeilberg/tokhniot/NCFPS
#
# This is slightly modified version of Prof. D. Zeilberger's
# NCFPS package. A few extra procedures added to suit our
# purpose.
#
# The internal representation of noncommutative polynomials
# is done using lists, same as in NCFPS package.
#
# Provides
# ========
#
# (F,G: noncommutative polynomials,
#  c: constnat,
#  m: monomial,
#  t: term)
#
# simplifyPoly(F): Simplifies F by collecting terms with the
#   same monomial. Additionaly, the returned polynomial has
#   terms arranged in decreasing order of degree (number of
#   indeterminates in a monomial).
#
# addPoly(F,G): Adds F and G.
#
# subtractPoly(F,G): F-G.
#
# sMulPoly(c,F): c*F (scalar multiplication)
#
# multPoly(F,G): F.G (multiplication)
#
# coeffPoly(m,F): coefficient of m in F.
#
```

```
# subsPoly(S,F): simplifies the coeffs in F using substitution
#   rules S (using the Maple builtin function "subs").
#   Example:
#     F := [[w^3, [x]]] (eqv. to (w^3)*x),
#     S := w^6=1
#     subscPoly(S,F) = [[-1,[x]]] (eqv. to -x).
#
# evalcPoly(F): simplifies the coeffs in F using evalc().
#
# coeffPoly(m,P): Finds the coefficient of the monomial m
#   in the polynomial P. Returns 0 if the monomial is not present.
################################################################

# simplifyPoly(F): simplifies F.
# -- Slight modification of "Pashet" form NCFPS package.
# -- Returned polynomials has terms sorted by degree.
# (these modifications ensures that the order of the monomials
# is optimal for the type of substitution rules we have for
# our computations. For example, the substitution rules
# based on the Lie brackets reduce the number of variables
# when we commute them. Thus we want to straighten out
# the longest monomial first.)

simplifyPoly := proc(F)
  local t, m, md, i, T, T1, L, M, M1, C;

  # M: collection of monomials in F
  M := {seq(t[2], t in F)};
  for m in M do
    C[m] := 0;
  end do;

  for t in F do
    #collecting coeffs
    C[t[2]] := C[t[2]] + t[1];
  end do;

  # T: collection of nonzero terms
  T := [];
  for m in M do
    if C[m] <> 0 then
```

```
      T := [op(T), [C[m], m]];
    end if;
  end do;

  # Boundary case: T = []
  if T = [] then
    return [];
  end if;

  # Simplify the constants (they maybe symbolic)
  for t in T do
    C[t[2]] := simplify(C[t[2]]);
  end do;

  # M1: collection of monomials with nonzero coeff (sorted)
  M1 := [seq(t[2], t in T)];
  M1 := sort(M1);

  # T1: terms in T sorted by monomials
  T1 := [seq([C[m], m], m in M1)];

  # Sorting by degree:

  #md: max degree (md >= 0)
  md := max(seq(nops(t[2]), t in T1));

  # L[i]: nonzero terms of degree i
  for i from 0 to md do
    L[i] := [];
  end do;

  for t in T1 do
    L[nops(t[2])] := [op(L[nops(t[2])]), t];
  end do;

  # Keep this order for faster processing in our case
  return [seq(op(L[i]), i=md..0, -1)];
end proc;

################################################################
```

```
# sMulPoly(c,F): c*F (scalar multiplication)--simplified.
# -- Same as "sMul" from NCFPS.

sMulPoly := proc(c, F)
  local i;
  return simplifyPoly([seq([c*F[i][1], F[i][2]], i=1..nops(F))]);
end proc;

################################################################

# multPoly(F,G): F.G (multiplication)--simplified
# -- Same as "Mul" in NCFPS.

multPoly := proc(F, G)
  return simplifyPoly([seq(seq(
      [fx[1]*gx[1], [op(fx[2]), op(gx[2])]],
    fx in F), gx in G)
  ]);
end proc;


################################################################

# addPoly(F,G): F+G (addition)--simplified.
# -- Same as "Khaber" from NCFPS

addPoly := proc(F,G)
  return simplifyPoly([op(F), op(G)]);
end proc;

################################################################

# subtractPoly(F,G): F-G (subtraction)--simplified.

subtractPoly := proc(F,G)
  return addPoly(F, sMulPoly(-1,G));
end proc;

################################################################

# coeffPoly(m,F): coefficient of monomial m in polynomial F.
```

```
coeffPoly := proc(m,F)
  local t, F1 := simplifyPoly(F);
  for t in F1 do
    if t[2] = m then
      return t[1];
    end if;
  end do;
  return 0;
end proc;

#################################################################

# subscPoly(S,F): simplifies the coeffs in F using substitution
#   rules S (using "subs"), and evalc().
#   Example:
#     F := [[w^3, [x]]] (eqv. to (w^3)*x),
#     S := w^6=1
#     subscPoly(S,F) = [[-1,[x]]] (eqv. to -x).

subscPoly := proc(S,F)
  local t, P := [];
  for t in F do
    P := [op(P), [evalc(subs(S, t[1])), t[2]]];
  end do;
  return simplifyPoly(P);
end proc;

#################################################################

# evalcPoly(F): simplifies the coeffs in F using evalc().

evalcPoly := proc(F)
  local t, P := [];
  for t in F do
    P := [op(P), [evalc(t[1]), t[2]]];
  end do;
  return simplifyPoly(P);
end proc;

#################################################################
```

```
# coeffPoly(m,P): Finds the coefficient of the monomial m
#   in the polynomial P. Returns 0 if the monomial is not present.
# Example: coeffPoly([x,y], [[2,[x,y]], [1,[x]]]) = 2
#
coeffPoly := proc(m, P)
  local Q;
  Q := select(x->x[2]=m, P);
  Q := simplifyPoly(Q);
  if nops(Q) = 0 then
    return 0;
  elif nops(Q) > 1 then
    return FAIL;
  end if;
  return Q[1][1];
 end proc;
```

We present the program to apply substitution rules. The algorithm and the Maple codes are adapted from M. Russel's [Rus13]. We have modified his algorithm to allow infinite number of rules (over indexed alphabet). We have added a few other procedures to suit our purpose. We have desiged the implementation in such a way that the main procedure `subsRule()` does not have to be changed, if we decide to code the substitution rules in a different way.

Listing B.3: npolysubs.txt

```
###############################################################
#
# File: npolysubs.txt
#
# Rewriting Noncommutative Polynomials Using Substitution Rules
# ============================================================
#
# Author: Debajyoti Nandi
#
# This file codes rewriting noncommutative polynomials using
# substitution rules of monomials. For example, if we have
# the rule x.y.x=1, then x.x.y.x.y reduces to x.y.
#
# Rules
# =====
```

```
#
# Rules are of the form
#   Monomial --> Polynomial
#
# Assume that R is a rule (or a patterns of similar rules)
#   m --> P, where m is a monomial, and P is a polynomial.
#
# Rules are coded (implemented) as a triplet:
#   R := [len, find(), substt()],
# where:
#   len: the length (or degree) of the monomial m;
#   find(F): finds first place matching the monomial m in the
#     polynomial F;
#   substt(m): returns the RHS P, (m assumed be the LHS of R).
#
# Special thanks to Matthew Russels for pointing out NCFPS
# package, and getting me started.
#
# Requires: npolyio.txt.
#
# Provides
# ========
#
# rewritePoly(F,Rules): Rewrites the polynomial F, using
#   rules in the set (or list) of rules in Rules. If this
#   procedure terminates, then it returns a reduced polynomial
#   where no more matching rules in Rules applies. There is
#   no guarantee that this will terminate, in case there are
#   cyclical substitutions possible with the rules.
#
# lenRule(R): returns the length of the monomial on the LHS
#   of the rule R.
#
# findMatchRule(m,R): finds the first place in the monomial m,
#   where the rule R applies.
#
# subsRule(m,R): returns the RHS (polynomial) of the rule R.
#
# Note
# ====
#
```

```
# Note monomials may be composed of indexed variables
# (infinitely many indeterminates), but rules should be
# described by finitely many patterns. See the file
# "A22-rules.txt" for examples.
#
# The last three procedures are used so that rewritePoly(),
# does not have to be changed, if one decides to re-implement
# rules. In that case, only the last three auxiliary
# procedures need to be modified.
#
################################################################

# lenRule(R): degree of the monomial on the LHS of the rule R.
lenRule := R -> R[1];

# findMatchRule(m,R): finds the first place in the monomial m,
# where the rule R applies.
findMatchRule := (m,R) -> R[2](m);

# subsRule(m,R): returns the RHS of the rule (m --> P).
subsRule := (m,R) -> R[3](m);

################################################################

rewritePoly := proc(F,Rules)
  local AllDone, NotDone, found, R, pre, suf, m, G, H, t, i;
  AllDone := []; #terms that are straightened
  NotDone := simplifyPoly(F); #terms that are not yet straightened

  while NotDone <> [] do
    t := NotDone[1];
    H := [];
    found := false; # no matching rule found

    for R in Rules while not found do
      i := findMatchRule(t[2], R);
      if i > 0 then
        found := true;
        pre := t[2][1..i-1];
        m := t[2][i..lenRule(R)+i-1];
        suf := t[2][lenRule(R)+i..];
```

```
        G := subsRule(m,R);
        H := multPoly(multPoly([[t[1],pre]], G), [[1,suf]]);
      end if;
    end do;

    NotDone := simplifyPoly([op(2.., NotDone), op(H)]);
    if not found then
      AllDone := [op(AllDone), t];
    end if;
  end do;

  AllDone := simplifyPoly(AllDone);
  return simplifyPoly(AllDone);
end proc;
```

## B.4 Substitution Rules

In this section, we present our Maple source files implementing the substitution rules that we require for our computations. These rules are split into two files: `A22-rules.txt` and `A22-L4-iniRules.txt`

These rules (except for the substitution rules coming from the initial conditions of Chapter 7) are presented in the Maple file below. The rules are divided into six sections in the Maple file. See the documentation in the Maple file below for the description of these rules.

Listing B.4: A22-rules.txt

```
################################################################
#
# File: A22-rules.txt
#
# Author: Debajyoti Nandi
#
# This files includes all rules needed for computations
# in the algebra $A_2^(2)$, or in their level 4 standard
# modules.
#
# Note:
#   w: 6th primitive root of unity,
```

```
#   w0: 12 the primitive root of unity (s.t. w0^2 = w).
# In this file their values are not set.
#
# Types of Rules
# ==============
#
# (Sec A): Rules given by the mapping of the Chevalley
#    generators in terms of vertex operators.
# Risom = the set of rules in Sec A.
#
# (Sec B): Bracket rules of the vertex operators.
# Rvop = the set of rules in Sec B.
#
# (Sec C): Bracket rules of the chevalley generators.
# Rgen = the set of rules in Sec C.
#
# (Sec D): Rules for all std modules, given by, positive degree
#    elements of A2(2) annihilates the highest weight vector
#    v[0] (i.e., e[i].v[0] = 0 for i=0,1).
#  We don't need the corresponding rules in terms of the
#  vertex operators, since we will only use negative
#  degree operators (when applying in terms of the vertex
#  operators) on v[0].
#
# (Sec E): Rules spicific to level 4 standard modules (i.e.,
#    h[0].v[0] = k0*v[0], h[1].v[0] = k1*v[0] in the
#    (k0,k1)-module). Note that the conditions
#    (f[0]^(k0+1)).v[0] = (f[1]^(k1+1)).v[0] = 0
#    follows from the action of h on v[0] and the Lie
#    brackets of the Chevalley generators.
# -- (Sec E40): for (4,0)-module;
# -- (Sec E21): for (2,1)-module;
# -- (Sec E02): for (0,2)-module.
# These rules are also in terms of the Chevalley generators only.
#
# RgenV<k0,k1>: The set of rules for the (k0,k1)-modules
#   in terms of the Chevalley generators. These rules
#   enables us to express any vector in V as a
#   linear combination of vectors of the form
#     f[n1]..f[nk].v[0]; n[1],...,n[k] = 0,1.
#   These rules are union of the rules in Sec C, Sec D, and
```

```
#   Sec E<k0,k1>.
#
# Auxiliary data/functions:
# -- A22: generalized Cartan Matrix for A2(2);
# -- delta(i,j): delta function;
# -- d6(i): 1 (if i congruent to 1,-1 mod 6), 0 (otherwise).
# -- omega(i,j): (w^2/6)*(w^(j-i) - w^(i-j))
#
# Note: (1) We use the rules "Risom" and "Rvop" to check the
# isomorphism of $A2(2)$,in terms of the vertex operatros.
#   (2) We use rules "Rgen", "RgenV<k0,k1>" and "Rvop" for our
# investigation of initial conditions in various level 4
# standard modules.

############################ Sec: A ############################

# (Sec A): Rules given by the mapping of the Chevalley
#    generators in terms of the vertex operators.

# rh0: h[0] --> (4*sqrt(3)/w0)*X[0] + (2/3)*c
rh0 := [
  1, #length

  proc(m) #find in m
    local i;
    for i from 1 to nops(m) do
      if m[i] = h[0] then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> [[4*sqrt(3)/w0,[X[0]]], [2/3,[c]]]
];

# rh1: h[1] --> (-2*sqrt(3)/w0)*X[0] + (1/6)*c
rh1 := [
  1, #length
```

```
  proc(m) #find h[1] in m
    local i;
    for i from 1 to nops(m) do
      if m[i] = h[1] then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> [[-2*sqrt(3)/w0,[X[0]]], [1/6,[c]]]
];

# re0: e[0] --> (-2*sqrt(2)/w0)*X[1] + (2*sqrt(2)/sqrt(3))*a[1]
re0 := [
  1, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m) do
      if m[i] = e[0] then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> [[-2*sqrt(2)/w0,[X[1]]], [2*sqrt(2)/sqrt(3),[a[1]]]]
];

# re1: e[1] --> (2/w0)*X[1] + (1/sqrt(3))*a[1]
re1 := [
  1, #length

  proc(m) #find match
    local i;
    for i from 1 to nops(m) do
      if m[i] = e[1] then
        return i;
```

```
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> [[2/w0,[X[1]]], [1/sqrt(3),[a[1]]]]
];

# rf0: f[0] --> (2*sqrt(2)/w0)*X[-1] + (2*sqrt(2)/sqrt(3))*a[-1]
rf0 := [
  1, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m) do
      if m[i] = f[0] then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> [[2*sqrt(2)/w0,[X[-1]]], [2*sqrt(2)/sqrt(3),[a[-1]]]]
];

# rf1: f[1] -> (-2/w0)*X[-1] + (1/sqrt(3))*a[-1]
rf1 := [
  1, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m) do
      if m[i] = f[1] then
        return i;
      end if;
    end do;
    return 0;
  end proc,
```

```
  #substt
  m -> [[-2/w0,[X[-1]]], [1/sqrt(3),[a[-1]]]]
];

Risom := {rh0,rh1,re0,re1,rf0,rf1};

############################ Sec: B #############################

# (Sec B): Bracket rules of the vertex operators.

d6 := i -> if i mod 6 in {1,5} then 1 else 0 end if;

omega := (i,j) -> if j-i mod 6 in {0,3} then 0
                  elif j-i mod 6 in {1,2} then -w0*sqrt(3)/6
                  else w0*sqrt(3)/6
                  end if;

# rcx: c.x --> x.c (x=X[*] or a[*])
rcx := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if m[i]=c and (op(0,m[i+1])=a or op(0,m[i+1])=X) then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> [[1,[m[2],m[1]]]]
];

# raa: a[i].a[j] --> a[j].a[i] + (delta(i+j,0)*i/6)*c, i>j
raa := [
  2, #length

  proc(m) #find
    local i;
```

```
    for i from 1 to nops(m)-1 do
      if op(0,m[i])=a and op(0,m[i+1])=a and
      op(1,m[i]) > op(1,m[i+1]) then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> simplifyPoly([[1,[m[2],m[1]]],
    [delta(op(1,m[1])+op(1,m[2]),0)*op(1,m[1])/6,[c]]])
];

# raX: a[i].X[j] --> X[j].a[i] + d6(i)*X[i+j], i>0
raX := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if op(0,m[i])=a and op(0,m[i+1])=X and op(1,m[i])>0 then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> simplify([[1,[m[2],m[1]]],
    [d6(op(1,m[1])), [X[op(1,m[1])+op(1,m[2])]]]])
];

# rXa: X[i].a[j] --> a[j].X[i] - d6(j)*X[i+j], j<0
rXa := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if op(0,m[i])=X and op(0,m[i+1])=a and op(1,m[i+1])<0 then
```

```
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m ->  simplify([[1,[m[2],m[1]]],
    [-d6(op(1,m[2])), [X[op(1,m[1])+op(1,m[2])]]]])
];

# rXX: X[i].X[j] --> X[j].X[i] + omega(i,j)*X[i+j]
#                     -((-1)^i*d6(i+j)*w/6)*a[i+j]
#                     + ((-1)^i*i*delta(i+j,0)*w/36)*c, i>j
rXX := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if op(0,m[i])=X and op(0,m[i+1])=X and
      op(1,m[i]) > op(1,m[i+1]) then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m ->  simplify([[1,[m[2],m[1]]],
    [omega(op(1,m[1]),op(1,m[2])),[X[op(1,m[1])+op(1,m[2])]]],
    [-(-1)^op(1,m[1])*d6(op(1,m[1])+op(1,m[2]))*w/6,
      [a[op(1,m[1])+op(1,m[2])]]],
    [(-1)^op(1,m[1])*op(1,m[1])*delta(op(1,m[1])+op(1,m[2]),0)*
      w/36, [c]]])
];

Rvop := {rcx,raa,raX,rXa,rXX};

############################ Sec: C ############################
```

```
# (Sec C): Bracket rules of the Chevalley generators.

# Generalized Cartan's Matrix for A2(2)
A22 := table([(0,0)=2, (0,1)=-4, (1,0)=-1, (1,1)=2]);

# delta function
delta := (i,j) -> if i=j then 1 else 0 end if;

# rhh: h[0].h[1] --> h[1].h[0]
rhh := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if m[i..i+1] = [h[0],h[1]] then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> [[1,[h[1],h[0]]]]
];

# rhe: h[i].e[j] --> e[j].h[i] + a22[i,j]*e[j], i,j=0,1
rhe := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if op(0,m[i])=h and op(0,m[i+1])=e then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
```

```
  m -> [[1,[m[2],m[1]]], [A22[op(1,m[1]),op(1,m[2])],[m[2]]]]
];

# rhf: h[i].f[j] --> f[j].h[i] - A22[i,j]*f[j], i,j=0,1
rhf := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if op(0,m[i])=h and op(0,m[i+1])=f then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> [[1,[m[2],m[1]]], [-A22[op(1,m[1]),op(1,m[2])],[m[2]]]]
];

# ref: e[i].f[j] --> f[j].e[i] + delta(i,j)*h[i]
ref := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if op(0,m[i])=e and op(0,m[i+1])=f then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> simplifyPoly([[1,[m[2],m[1]]],
        [delta(op(1,m[1]),op(1,m[2])),[h[op(1,m[1])]]]])
];

Rgen := {rhh,rhe,rhf,ref};
```

```
############################ Sec: D #############################

# (Sec D): Rules for all std modules, given by, positive degree
#     elements of A2(2) annihilates the highest weight vector
#     v[0] (i.e., e[i].v[0] = 0 for i=0,1).
#     We don't need the corresponding rules in terms of the
#     vertex operators, since we will only use negative
#     degree operators (when applying in terms of the vertex
#     operators) on v[0].

# rev: e[*].v[0] --> 0
rev := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if op(0,m[i])=e and m[i+1]=v[0] then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> []
];

############################ Sec: E #############################

# (Sec E): Rules spicific to level 4 standard modules (i.e.,
#     h[0].v[0] = k0*v[0], h[1].v[0] = k1*v[0] in the
#     (k0,k1)-module). Note that the conditions
#     (f[0]^(k0+1)).v[0] = (f[1]^(k1+1)).v[0] = 0
#     follows from the action of h on v[0] and the Lie
#     brackets of the Chevalley generators.

### (Sec E40): for (4,0)-module;

# rh0v40: h[0].v[0] --> 4*v[0]
```

```
rh0v40 := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if m[i..i+1] = [h[0],v[0]] then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> [[4,[v[0]]]]
];

# rh1v40: h[1].v[0] --> 0
rh1v40 := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if m[i..i+1] = [h[1],v[0]] then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> []
];

RgenV40 := Rgen union {rev, rh0v40, rh1v40};

### (Sec E21): for (2,1)-module;

# rh0v21: h[0].v[0] --> 2*v[0]
rh0v21 := [
```

```
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if m[i..i+1] = [h[0],v[0]] then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> [[2,[v[0]]]]
];

# rh1v21: h[0].v[0] --> v[0]
rh1v21 := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if m[i..i+1] = [h[1],v[0]] then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> [[1,[v[0]]]]
];

RgenV21 := Rgen union {rev, rh0v21, rh1v21};

### (Sec E02): for (0,2)-module;

# rh0v02: h[0].v[0] --> 0
rh0v02 := [
  2, #length
```

```
  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if m[i..i+1] = [h[0],v[0]] then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> []
];

# rh1v02: h[1].v[0] --> 2*v[0]
rh1v02 := [
  2, #length

  proc(m) #find
    local i;
    for i from 1 to nops(m)-1 do
      if m[i..i+1] = [h[1],v[0]] then
        return i;
      end if;
    end do;
    return 0;
  end proc,

  #substt
  m -> [[2,[v[0]]]]
];

RgenV02 := Rgen union {rev, rh0v02, rh1v02};
```

The following file contains the replacement rules coming from the initial conditions (as described in the proofs of Chapter 7).

Listing B.5: A22-L4-iniRules.txt

```
################################################################
#
```

```
# File: A22-L4-iniRules.txt
#
# Author: Debajyoti Nandi
#
# This file contains the replacement rules coming
# from the initial conditions as described in the proofs
# of Chapter 7 (Initial Conditions).
#
################################################################

#################### For the (4,0)-module ####################
#
# rini40a: Coming from f[1].v[0] = 0 in (4,0)-module.
# X[-1].v[0] --> (w0*sqrt(3)/6)*a[-1].v[0]
#
rini40a := [
  2, #length

  proc(m) #find
    if m[-2..] = [X[-1],v[0]] then
      return nops(m)-1;
    end if;
    return 0;
  end proc,

  #substt
  m -> [[w0*sqrt(3)/6,[a[-1],v[0]]]]
];

#################### For the (2,1)-module ####################
#
# rini21a: Coming from (f[1]^2).v[0] = 0 in (2,1)-module
# X[-1].X[-1].v[0] --> (-w0*sqrt(3)/6)*X[-2].v[0]
#  + (w0*sqrt(3)/3)*a[-1].X[-1].v[0]
#  + (-w/12)*a[-1].a[-1].v[0]
#
rini21a := [
  3, #length

  proc(m) #find
    if m[-3..] = [X[-1],X[-1],v[0]] then
```

```
      return nops(m)-2;
    end if;
    return 0;
  end proc,

  #substt
  m -> [[-w0*sqrt(3)/6, [X[-2],v[0]]],
        [w0/sqrt(3), [a[-1],X[-1],v[0]]],
        [-w/12, [a[-1],a[-1],v[0]]]]
];

# rini21b: Coming from (f[0]^3).v[0] = 0 in (2,1)-module.
# X[-2].X[-1].v[0] --> (-w0/sqrt(3))*a[-1].X[-2].v[0]
#               + (w0*sqrt(3)/2)*a[-1].a[-1].X[-1].v[0]
#
rini21b := [
  3, #length

  proc(m) #find
    if m[-3..] = [X[-2],X[-1],v[0]] then
      return nops(m)-2;
    end if;
    return 0;
  end proc,

  #substt
  m -> [[-w0/sqrt(3), [a[-1],X[-2],v[0]]],
        [w0*sqrt(3)/2, [a[-1],a[-1],X[-1],v[0]]]]
];

# rini21c: Coming from (f[0]^4).v[0] = 0 in (2,1)-module.
# (X[-2])^2.v[0] --> (-4/3)*X[-3].X[-1].v[0]
#  + (-w0*sqrt(3)/18)*X[-4].v[0]
#  + (w0*2*sqrt(3)/9)*a[-1].X[-3].v[0]
#  + (-w0*sqrt(3))*(a[-1]^2).X[-2].v[0]
#  + (w0*4/sqrt(3))*(a[-1]^3).X[-1].v[0] + (w/12)*(a[-1]^4).v[0]
#
rini21c := [
  3, #length

  proc(m) #find
```

```
    if m[-3..] = [X[-2],X[-2],v[0]] then
      return nops(m)-2;
    end if;
    return 0;
  end proc,

  #substt
  m -> [[-4/3, [X[-3],X[-1],v[0]]],
        [-w0*sqrt(3)/18, [X[-4],v[0]]],
        [w0*2*sqrt(3)/9, [a[-1],X[-3],v[0]]],
        [-w0*sqrt(3), [a[-1]$2,X[-2],v[0]]],
        [w0*4/sqrt(3), [a[-1]$3,X[-1],v[0]]],
        [w/12, [a[-1]$4,v[0]]]]
];

# rini21d: coming from (f[0]^5).v[0] = 0 in (2,1)-module.
#   Re-write rule for X[-3].X[-2].v[0] ...
#
rini21d := [
  3, #length

  proc(m) #find
    if m[-3..] = [X[-3],X[-2],v[0]] then
      return nops(m)-2;
    end if;
    return 0;
  end proc,

  #substt
  m -> [[-1, [X[-4],X[-1],v[0]]], [1, [a[-1],X[-3],X[-1],v[0]]],
        [-w0*sqrt(3)/6, [a[-1],X[-4],v[0]]],
        [w0/sqrt(3), [a[-1]$2,X[-3],v[0]]],
        [-w0*sqrt(3)/2, [a[-1]$3,X[-2],v[0]]],
        [w0*sqrt(3)/2, [a[-1]$4,X[-1],v[0]]],
        [w/20, [a[-1]$5,v[0]]], [w/30, [a[-5],v[0]]]]
];

# rini21e: Coming from (f[0]^6).v[0] = 0 in (2,1)-module
# Replacement rule for X[-3].X[-3].v[0] --> ...
#
rini21e := [
```

```
  3, #length

  proc(m) #find
    if m[-3..] = [X[-3],X[-3],v[0]] then
      return nops(m)-2;
    end if;
    return 0;
  end proc,

  #substt
  m -> [[2, [X[-5],X[-1],v[0]]], [-w0*sqrt(3)/6, [X[-6],v[0]]],
        [-6, [a[-1],X[-4],X[-1],v[0]]],
        [w0*2/sqrt(3), [a[-1],X[-5],v[0]]],
        [6, [a[-1]$2,X[-3],X[-1],v[0]]],
        [-w0*sqrt(3), [a[-1]$2,X[-4],v[0]]],
        [w0*sqrt(3), [a[-1]$3,X[-3],v[0]]],
        [-w0*3*sqrt(3)/5, [a[-1]$5,X[-1],v[0]]],
        [-w0*sqrt(3)/15, [a[-5],X[-1],v[0]]],
        [-w/6, [a[-5],a[-1],v[0]]]]
];


#################### For the (0,2)-module ####################
#
# rini02a: Coming from f[0].v[0] = 0 in (0,2)-module
# Replacement rule for X[-1].v[0] --> ...
#
rini02a := [
  2, #length

  proc(m) #find
    if m[-2..] = [X[-1],v[0]] then
      return nops(m)-1;
    end if;
    return 0;
  end proc,

  #substt
  m -> [[-w0/sqrt(3), [a[-1],v[0]]]]
];
```

```
# rini02b: Coming from (f[1]^3).v[0] = 0 in (0,2)-module
# Replacement rule for X[-3].v[0] --> ...
#
rini02b := [
  2, #length

  proc(m) #find
    if m[-2..] = [X[-3],v[0]] then
      return nops(m)-1;
    end if;
    return 0;
  end proc,

  #substt
  m -> [[3/2, [a[-1],X[-2],v[0]]],
        [w0*sqrt(3)/4, [a[-1]$3,v[0]]]]
];

# rini02c: Coming from (f[1]^4).v[0] = 0 in (0,2)-module
# Replacement rule for (X[-2]^2).v[0] --> ...
rini02c := [
  3, #length

  proc(m) #find
    if m[-3..] = [X[-2],X[-2],v[0]] then
      return nops(m)-2;
    end if;
    return 0;
  end proc,

  #substt
  m -> [[-w0*sqrt(3)/2, [X[-4],v[0]]],
        [w0*sqrt(3), [a[-1]$2,X[-2],v[0]]],
        [w*3/4, [a[-1]$4,v[0]]]]
];
```

## B.5 Other Miscellaneous Maple Codes Used

In this section, we present the auxiliary Maple source file `misc.txt` which includes miscellaneous procedures used in our Maple worksheets.

Listing B.6: misc.txt

```
#################################################################
#
# File: misc.txt
#
# Author: Debajyoti Nandi
#
# Miscellaneous Procedures Used Elsewhere
# =======================================
#
# In this file, we list a few procedures used in the Maple
# worksheets else where.
#
# Provides:
# =========
# b(F, G): Lie bracket of F, G (noncommutative polynomials)
#
# adpow(k, F, G): ((ad F)^k).G
#
# is0inV(u, T): checks if the monomial (f[0], f[1]) acting
#   v[0] is 0 in the std module V of type T, where
#   T = "40", "21" or "02".
#
# genseq(k): generates all binary sequences of length k.
#
# genF(s): given a binary sequence s, it maps the sequence
#   into the monomial in f[0] and f[1] acting on v[0].
#
#################################################################

# b(F,G): [F,G] (Lie bracket)
b := (F,G) -> subtractPoly(multPoly(F,G), multPoly(G,F));

# adpow(k,F,G): (ad F)^k.G
adpow := proc(k,F,G)
  local i, H := G;
  for i from 1 to k do
    H := b(F,H);
  end do;
  return H;
end proc;
```

```
# is0inV(u,T): Checks if a homogeneous vector u of negative
#   degree (user's responsibility to enforce this)
#   in V (std module of type T, T="40", "21" or "02")
#   by checking if e[0].v = e[1].v = 0. Here, v is
#   assumed to be in terms of the Chevalley generators
#   acting on the highest weight vector v[0].
#
# Requires: npolyops.txt, npolysubs.txt, A22-rules.txt
#
is0inV := proc(u,T)
  local U := simplifyPoly(u), Rules := RgenV||T;

  #boundary checks
  if nops(U)=1 and U[1][2] = [v[0]] then
    return false;
  elif U = [] then
    return true;
  end if;

  #recursive checks
  if is0inV(rewritePoly(multPoly([[1,[e[0]]]],U),Rules),T) and
  is0inV(rewritePoly(multPoly([[1,[e[1]]]],U),Rules),T) then
    return true;
  end if;

  return false;
end proc;

# Generates binary sequences of length k
# Example: genseq(3) = [[0, 0], [0, 1], [1, 0], [1, 1]]
#
genseq := proc(k)
  if k=0 then return [[]] end if;
  return [seq(seq([i,op(L)], L in genseq(k-1)), i=0..1)];
end proc;

# Converts a binary seq [i1,...,in] into f[i1]...f[in].v[0]
# (as a noncommutative polynomial using list)
# Example: genF([1,0]) = [[1,[f[1],f[0],v[0]]]]
#
```

```
genF := proc(sq)
  local s, m := [];
  for s in sq do
    m := [op(m), f[s]];
  end do;
  return [[1, [op(m),v[0]]]];
end proc;
```

# Appendix C

# Verification of the Partition Identities

In this appendix, we present the C program that we used to verify the three partition identities presented in Chapter 8. We have verified the results upto $n \leq 170$, and for $n = 180, 190$ and 200. (Note that the computation for $n = 200$, may take more than 24 hours to complete).

We used Kelleher's algorithm from [Kel06] to generate all partitions of $n$. We used the accelerated ascending rule algorithm. This algorithm produces partitions as a non-decreasing list. When a partition is generated we filter it out based on the criteria presented in Chapter 8.

The program is split into two files. In § C.1, we present the main file (written in C with C99 standard) `verify.c` implementing Kelleher's accelerated ascending rule algorithm to generate partitions, and our "visitor" function to check if the generated partition should be counted for each of the standard $\mathrm{A}_2^{(2)}$-modules of level 4 (based on the criteria presented in Chapter 8).

In § C.2, we list the file `data.h`, which contains data about $\mathscr{P}(n)$ (see Notation 4.1.7) and $F_\Lambda(n)$ (see § 3.2, Notation 3.2.4) for $0 \leq n \leq 200$, where $\Lambda = 4h_0^*$, $2h_0^* + h_1^*$ or $2h_1^*$. These numbers were generated using power series expansion (using the Maple package `powseries`) of the product side $F_\Lambda(q)$ (Notation 3.2.4).

## C.1 File: `verify.c`

In this section, we present the main file (written in C with C99 standard) `verify.c` implementing Kelleher's accelerated ascending rule algorithm to generate partitions, and our "visitor" function to check if the generated partition should be counted for each of the standard $\mathrm{A}_2^{(2)}$-modules of level 4 (based on the criteria presented in Chapter 8).

To compile save this file and the auxiliary file `data.h` from § C.2 in the same working directory. To compile and run on a GNU/Linux machine, use the following commands.

```
$ cc -std=c99 verify.c -o verify
$ ./verify
```

Listing C.1: verify.c

```
/************************************************************\

File: verify.c
Author: Debajyoti Nandi
Email: nandi@math.rutgers.edu

Description:
  Verifies the three partition identities (up to n <= 200).
  The program implements Kelleher's accelAscRule algorithm
  to generate partitions. Once a partition is generated,
  the function fltrCnt() then checks to see if the partition
  is allowed in each of the  three level 4 standard modules,
  and counts.

Note(1):
  To compile with cc and run
    $ cc -std=c99 verify.c -o verify
    $ ./verify

  Enter the min and max of the range over which to check,
  (min <= n <= max).

Note(2):
  Compile with -DPRINT flag to display the "allowed"
  partitions.

Warning: The above is only useful for debugging with
  small values of n. Otherwise, the output will be
  overwhelmingly verbose.

\************************************************************/

#include <stdio.h>
#include <stdlib.h>
```

```
#include "data.h"

/*
 * visitor function type to be called after a partition
 * has been generated.
 */
typedef void (*Visitor) (int p[], int k, long *counts);

/*
 * Kelleher's [Kel06] accelerated ascending rule algorithm
 * to generate partitions of 'n' > 0. The visitor
 * function 'f' is called once a partition is generated.
 * 'counts' is an array, used by 'f' to counts the number
 * of "allowed" partitions in each of the three cases.
 */
long long accelAsc(int n, Visitor f, long *counts);

/*
 * Prints the partition (which is produced as a
 * non-decreasing sequence) in reverse order, ie,
 * in the non-increasing order. 'k' is the length
 * of the generated partition.
 */
void printRev(int p[], int k);

/*
 * 'fltrCnt()' is the visitor function implementation
 * to check if the partitions are "allowed" or not
 * for each of the three cases. If a partition is allowed,
 * it is then counted for the appropriate module.
 */
void fltrCnt(int p[], int k, long *counts);

/* Various states, used in the definition of fltrCnt() */
enum states_fltrCnt {
  F40 = 1,  /* reducible for (4,0)-module */
  F21 = 2,  /* reducible for (2,1)-module */
  F02 = 4,  /* reducible for (0,2)-module */
  FL4 = 7,  /* reducible for all level 4 modules */
  S03 = 8,  /* entered [2*,3,0] diff condn */
  S2 = 16,  /* entered [2*] diff condn starting at 2 */
```

```
  ODD = 32, /* current entry is odd */
};


int main(int argc, char *argv[]) {
  int min, max;
  long cnts[3];
  long long count;

  printf("Starting from: ");
  scanf("%d", &min);
  printf("Ending at: ");
  scanf("%d", &max);
  for (int n=min; n<=max; n++) {
    printf("Computing Partitions of n=%d...\n", n);
    count = accelAsc(n, fltrCnt, cnts);
    if (X[n] != count) printf("**");
    printf("\tAll=%13lld\tGot=%13lld\n", X[n], count);
    if (X40[n] != cnts[0]) printf("**");
    printf("\tX40=%13ld\tGot=%13ld\n", X40[n], cnts[0]);
    if (X21[n] != cnts[1]) printf("**");
    printf("\tX21=%13ld\tGot=%13ld\n", X21[n], cnts[1]);
    if (X02[n] != cnts[2]) printf("**");
    printf("\tX02=%13ld\tGot=%13ld\n", X02[n], cnts[2]);
    printf("\n");
  }
  return 0;
}


void printRev(int a[], int k) {
  int i;
  printf("[");
  for (i=0; i<k; i++) {
    if (i>0) {
      printf(",");
    }
    printf("%d", a[k-i-1]);
  }
  printf("]\n");
}
```

```
long long accelAsc(int n, Visitor f, long *counts) {
  int k, j, i, x, y;
  int a[n];
  long long count=0;
  counts[0] = counts[1] = counts[2] = 0;
  if (n == 0) {
    f(a, 0, counts); /* null partition */
    return 1;
  }
  for (i=0; i<n; i++) {
    a[i] = 0;
  }
  k = 1;
  y = n-1;
  while (k != 0) {
    x = a[k-1] + 1;
    k--;
    while (2*x <= y) {
      a[k] = x;
      y = y - x;
      k++;
    }
    j = k + 1;
    while (x <= y) {
      a[k] = x;
      a[j] = y;
      f(a, k+2, counts); /* generated a partition */
      count++;
      x++;
      y--;
    }
    a[k] = x + y;
    y = x + y - 1;
    f(a, k+1, counts); /* generated a partition */
    count++;
  }
  return count;
} /* End of accelAsc() */
```

```
void fltrCnt(int p[], int k, long *counts) {
  /*
  Partitions are represented as non-decreasing lists.
  (2,5,6) <=> (6,5,2) read R to L
    if currently reading the 3rd index (6), then
      cur = 6
      prv = 5
      d = 1
      d1 = 3

  States:
    ODD => Whether cur is odd or not
    S03 => current diff condn is [2*,3,0]
    S2  => current diff condn is [2*] starting at 2
    F40 => partition is reducible for (4,0)-module
    F21 => partition is reducible for (2,1)-module
    F02 => partition is reducible for (0,2)-module
    FL4 => partition is reducible for all level 4 modules.

  IC:
    (1) => sets F40 & F02
    (3) => sets F02
    (1,1), (3,3) => sets F21
    (2,2) => sets F21 & F02
    [3-,2*] starting with 2 => sets F02

  ALL:
    [1],
    [0,0], [0,2], [2,0], [0,3],
    [3-,0], [0-,4], [4-,0],
    [3-,2*,3,0]
      => sets FL4 (ie, returns)

  States switching:
    S03:
      sets when diff conds reaches [3,0],
      resets when
        d == 3 and cur is ODD => sets FL4
        otherwise if d != 2.
```

```
  S02:
    sets when the first entry is 2,
    resets when
      d == 3 (no need to check ODD) => sets F02
      otherwise if d != 2.

  ODD:
    sets when cur is odd,
    resets when cur is even.
*/
int cur;      /* current part being read */
int prv;      /* previous part read */
int d;        /* difference = cur - prv */
int d1 = -1; /* last difference */
char s = 0;  /* Bits of "s" represent different states */

for (int i=0; i<k; i++) {
  cur = p[i];
  if (cur % 2)
    s |= ODD;
  else
    s &= ~ODD;
  if (i == 0) { /* we are reading the 1st entry */
    switch(cur) {
      case 1:
        s |= (F40|F02);
        break;
      case 2:
        s |= S2;
        break;
      case 3:
        s |= F02;
        break;
      default:
        break;
    }
  } else { /* reading 2nd or further to the left */
    d = cur - prv;
    if ((d==1) || (d==0 && d1==0) || (d==0 && d1==2)
        || (d==2 && d1==0) || (d==0 && d1==3)
        || ((s & ODD) && d==3 && d1==0)
```

```
        || ((s & ODD) && d==0 && d1==4)
        || ((s & ODD) && d==4 && d1==0)
        || ((s & S03) && d==3 && (s & ODD)))
      return;
    if (s & S2) {
      switch (d) {
        case 2:
          break;
        case 3:
          s |= F02;
          if ((s & FL4)==FL4) return;
          s &= ~S2;
          break;
        default:
          s &= ~S2;
          break;
      }
    }
    if ((s & S03) && d != 2)
      s &= ~S03;
    else if (d==3 && d1==0)
      s |= S03;
    if ((i==1)&&(cur==prv)&&(cur==1 || cur==3)) {
      s |= F21;
      if ((s & FL4)==FL4) return;
    } else if ((i==1)&&(cur==prv)&&(cur==2)) {
      s |= (F21|F02);
      if ((s & FL4)==FL4) return;
    }
    d1 = d;
  } /* End of if (i==0) */
  prv = cur;
} /* End of for(i) */
if (!(s & F40)) {
  counts[0]++;
  #ifdef PRINT
  printf("(4,0)-module: ");
  printRev(p, k);
  #endif
}
if (!(s & F21)) {
```

```
    counts[1]++;
    #ifdef PRINT
    printf("(2,1)-module: ");
    printRev(p, k);
    #endif
  }
  if (!(s & F02)) {
    counts[2]++;
    #ifdef PRINT
    printf("(0,2)-module: ");
    printRev(p, k);
    #endif
  }
} /* End of fltrCnt() */
```

## C.2 File: `data.h`

In this section, we present the file `data.h`, which contains data about $\mathscr{P}(n)$ (see Notation 4.1.7) and $F_\Lambda(n)$ (see § 3.2, Notation 3.2.4) for $0 \leq n \leq 200$, where $\Lambda = 4h_0^*$, $2h_0^* + h_1^*$ or $2h_1^*$. These numbers were generated using power series expansion (using the Maple package `powseries`) of the product side $F_\Lambda(q)$ (Notation 3.2.4).

Listing C.2: data.h

```
/**************************************************************\

File: data.h
Author: Debajyoti Nandi
Email: nandi@math.rutgers.edu

Description:
  Contains the data about the graded dimensions of the level 4
  standard modules for A2(2).

  XMN[n] = the coefficient of q^n in the second product
           expansion (F_{\Lambda}(q)) of the graded dimension
           of the (M,N)-module.

  MN = 40, 21 or 02.

```

```
  X40[n] is the number of partitions of n into
    parts congruent to +/-{2,3,4} modulo 14.

  X21[n] is the number of partitions of n into
    parts congruent to +/-{1,4,6} modulo 14.

  X02[n] is the number of partitions of n into
    parts congruent to +/-{2,5,6} modulo 14.

  X[n] is the number of all partitions of n.

Note:
  These numbers are directly computed from the expansion of
  the corresponding product formula (using Maple package
  "powseries").

  0 <= n <= MAX-1 (MAX=201 defined below).

  We checked the equality of the spanning set for n <= 170.

\***************************************************************/

#ifndef DATA_H
#define DATA_H

#define MAX 201 /* one plus the max */

/* Needs at least 32 bit */
const long X40[MAX] = {1, 0, 1, 1, 2, 1, 3, 2, 4, 3, 6, 5, 9, 7,
12, 11, 17, 15, 23, 21, 31, 29, 41, 39, 55, 52, 71, 70, 93, 91,
120, 119, 154, 154, 196, 198, 250, 252, 314, 321, 395, 404, 494,
508, 615, 635, 762, 790, 943, 978, 1159, 1209, 1423, 1485, 1740,
1821, 2121, 2224, 2577, 2708, 3126, 3286, 3776, 3980, 4554, 4802,
5477, 5783, 6571, 6945, 7865, 8321, 9397, 9945, 11197, 11865,
13320, 14118, 15812, 16770, 18735, 19879, 22155, 23520, 26159,
27774, 30824, 32746, 36268, 38532, 42601, 45273, 49961, 53104,
58501, 62193, 68407, 72724, 79863, 84922, 93117, 99012, 108418,
115289, 126066, 134057, 146394, 155676, 169796, 180546, 196682,
209140, 227565, 241953, 262984, 279596, 303570, 322717, 350025,
372071, 403164, 428496, 463857, 492964, 533149, 566517, 612163,
650394, 702190, 745936, 804669, 854678, 921244, 978328, 1053701,
```

```
1118837, 1204125, 1278328, 1374785, 1459264, 1568262, 1664337,
1787425, 1896597, 2035525, 2159436, 2316120, 2456690, 2633289,
2792562, 2991505, 3171854, 3395806, 3599839, 3851793, 4082447,
4365758, 4626262, 4944619, 5238677, 5596211, 5927823, 6329149,
6702882, 7153082, 7573964, 8078710, 8552370, 9117990, 9650628,
10284088, 10882725, 11591784, 12264060, 13057354, 13811892,
14698931, 15545230, 16536585, 17485229, 18592642, 19655289,
20891688, 22081420, 23461150, 24792263, 26331227, 27819720,
29535426, 31198920, 33110760, 34968805, 37098243, 39172364,
41543017, 43857204, 46495191, 49075706, 52009892, 54885951,
58148084};

const long X21[MAX] = {1, 1, 1, 1, 2, 2, 3, 3, 5, 5, 7, 7, 10,
11, 14, 15, 20, 21, 27, 29, 37, 40, 49, 53, 66, 71, 86, 93, 113,
122, 146, 158, 188, 204, 240, 260, 306, 332, 386, 419, 487, 528,
609, 661, 760, 825, 943, 1023, 1168, 1267, 1438, 1560, 1767,
1916, 2162, 2344, 2639, 2860, 3209, 3476, 3894, 4217, 4708,
5097, 5682, 6148, 6836, 7394, 8207, 8874, 9828, 10621, 11746,
12689, 14003, 15121, 16665, 17986, 19788, 21348, 23455, 25293,
27745, 29905, 32766, 35302, 38621, 41592, 45453, 48926, 53399,
57455, 62639, 67368, 73360, 78862, 85794, 92189, 100180, 107602,
116823, 125422, 136034, 145987, 158200, 169704, 183731, 197007,
213120, 228427, 246893, 264519, 285686, 305955, 330174, 353460,
381161, 407883, 439521, 470148, 506274, 541345, 582526, 622644,
669581, 715421, 768846, 821178, 881958, 941644, 1010713,
1078717, 1157172, 1234590, 1323598, 1411652, 1512596, 1612653,
1727012, 1840620, 1970107, 2098992, 2245472, 2391561, 2557186,
2722651, 2909739, 3097001, 3308240, 3520000, 3758301, 3997592,
4266275, 4536476, 4839180, 5144052, 5484912, 5828670, 6212188,
6599520, 7030829, 7466949, 7951648, 8442383, 8986794, 9538617,
10149685, 10769785, 11455329, 12151724, 12920294, 13701879,
14563118, 15439763, 16404234, 17386918, 18466449, 19567341,
20774941, 22007525, 23357735, 24736987, 26245747, 27788242,
29473390, 31197471, 33078570, 35004574, 37103408, 39253822,
41594347, 43994022, 46602870, 49279318, 52185710, 55169310,
58405745, 61730037, 65332175, 69034230, 73041596};

const long X02[MAX] = {1, 0, 1, 0, 1, 1, 2, 1, 3, 2, 4, 3, 6, 4,
8, 6, 11, 9, 15, 12, 20, 17, 26, 23, 35, 31, 45, 41, 58, 54, 75,
70, 96, 91, 121, 117, 154, 149, 193, 189, 242, 239, 302, 299,
375, 375, 463, 466, 572, 577, 702, 712, 859, 876, 1049, 1072,
```

```
1277, 1310, 1548, 1594, 1875, 1934, 2262, 2340, 2723, 2825,
3271, 3398, 3920, 4081, 4685, 4887, 5592, 5839, 6656, 6962,
7908, 8284, 9379, 9833, 11103, 11654, 13116, 13782, 15473,
16270, 18218, 19173, 21416, 22557, 25137, 26489, 29458, 31063,
34466, 36366, 40275, 42510, 46989, 49620, 54749, 57839, 63706,
67319, 74032, 78257, 85916, 90846, 99594, 105327, 115302,
121966, 133332, 141065, 154005, 162955, 177684, 188036,
204771, 216726, 235748, 249522, 271114, 286975, 311469, 329709,
357475, 378409, 409874, 433884, 469493, 496997, 537294, 568750,
614307, 650257, 701730, 742769, 800888, 847671, 913268, 966560,
1040520, 1101165, 1184537, 1253465, 1347369, 1425654, 1531362,
1620192, 1739117, 1839800, 1973535, 2087573, 2237838, 2366892,
2535676, 2681578, 2871034, 3035871, 3248420, 3434499, 3672822,
3882693, 4149796, 4386347, 4685500, 4951920, 5286847, 5586662,
5961417, 6298600, 6717710, 7096653, 7565130, 7990684, 8514112,
8991718, 9576197, 10111869, 10764269, 11364646, 12092472,
12764978, 13576575, 14329409, 15233989, 16076195, 17083983,
18025646, 19147834, 20200078, 21449166, 22624261, 24013932,
25325521, 26870925, 28334046, 30051881, 31683105, 33591834};

/* Needs at least 64 bit */
const long long X[MAX] = {1, 1, 2, 3, 5, 7, 11, 15, 22, 30, 42,
56, 77, 101, 135, 176, 231, 297, 385, 490, 627, 792, 1002, 1255,
1575, 1958, 2436, 3010, 3718, 4565, 5604, 6842, 8349, 10143,
12310, 14883, 17977, 21637, 26015, 31185, 37338, 44583, 53174,
63261, 75175, 89134, 105558, 124754, 147273, 173525, 204226,
239943, 281589, 329931, 386155, 451276, 526823, 614154, 715220,
831820, 966467, 1121505, 1300156, 1505499, 1741630, 2012558,
2323520, 2679689, 3087735, 3554345, 4087968, 4697205, 5392783,
6185689, 7089500, 8118264, 9289091, 10619863, 12132164, 13848650,
15796476, 18004327, 20506255, 23338469, 26543660, 30167357,
34262962, 38887673, 44108109, 49995925, 56634173, 64112359,
72533807, 82010177, 92669720, 104651419, 118114304, 133230930,
150198136, 169229875, 190569292, 214481126, 241265379, 271248950,
304801365, 342325709, 384276336, 431149389, 483502844, 541946240,
607163746, 679903203, 761002156, 851376628, 952050665,
1064144451, 1188908248, 1327710076, 1482074143, 1653668665,
1844349560, 2056148051, 2291320912, 2552338241, 2841940500,
3163127352, 3519222692, 3913864295, 4351078600, 4835271870,
5371315400, 5964539504, 6620830889, 7346629512, 8149040695,
9035836076, 10015581680, 11097645016, 12292341831, 13610949895,
```

```
15065878135, 16670689208, 18440293320, 20390982757, 22540654445,
24908858009, 27517052599, 30388671978, 33549419497, 37027355200,
40853235313, 45060624582, 49686288421, 54770336324, 60356673280,
66493182097, 73232243759, 80630964769, 88751778802, 97662728555,
107438159466, 118159068427, 129913904637, 142798995930,
156919475295, 172389800255, 189334822579, 207890420102,
228204732751, 250438925115, 274768617130, 301384802048,
330495499613, 362326859895, 397125074750, 435157697830,
476715857290, 522115831195, 571701605655, 625846753120,
684957390936, 749474411781, 819876908323, 896684817527,
980462880430, 1071823774337, 1171432692373, 1280011042268,
1398341745571, 1527273599625, 1667727404093, 1820701100652,
1987276856363, 2168627105469, 2366022741845, 2580840212973,
2814570987591, 3068829878530, 3345365983698, 3646072432125,
3972999029388};

#endif
```

# References


[And94] G. E. Andrews, "Schur's theorem, Capparelli's conjecture, and $q$-trinomial coefficients," in: *Proc. Rademacher Centenary Conf. (1992)*, Contemporary Math. **167**, Amer. Math. Soc., Providence, 1994, pp. 141–154.

[BRRZ12] A. Berenstein, V. Retakh, C. Reuternauer, and D. Zeilberger, "The reciprocal of $\sum_{n\geq 0} a^n b^n$ for non-commuting $a$ and $b$, Catalan numbers and non-commutative quadratic equations," *arXiv preprint arXiv:1206.4225* (2012), URL: `http://arxiv.org/abs/1206.4225`.

[Cap88] S. Capparelli, "Vertex operator relations for affine algebras and combinatorial identities," PhD thesis, Rutgers University, 1988.

[Cap92] S. Capparelli, "Elements of the annihilating ideal of a standard module," *J. Algebra* **145** (1 1992), pp. 32–54.

[Cap93] S. Capparelli, "On some representations of twisted affine Lie algebras and combinatorial identities," *J. Algebra* **154** (2 1993), pp. 335–355.

[Cap96] S. Capparelli, "A construction of the level 3 modules for the affine Lie algebra $\mathrm{A}_2^{(2)}$ and a new combinatorial identity of the Rogers-Ramanujan type," *Trans. Amer. Math. Soc.* **348**.2 (1996), pp. 481–501.

[Cha11] S.-H. Cha, *Recursive algorithm for generating partitions of an integer*, tech. rep. 280, Pace University, Seidenberg School of Computer Science and Information Systems, 2011.

[DL96] C. Dong and J. Lepowsky, "The algebraic structure of relative twisted vertex operators," *J. Pure Appl. Math.* **110** (3 1996), pp. 259–295, URL: `http://arxiv.org/abs/q-alg/9604022`.

[Fig87] L. Figueiredo, "Calculus of principally twisted vertex operators," *Mem. Amer. Math. Soc.* **371** (1987).

[FK80] I. B. Frenkel and V. G. Kac, "Basic representations of affine Lie algebras and dual resonance modules," *Invent. Math.* **62** (1980), pp. 23–66.

[FLM87] I. B. Frenkel, J. Lepowsky, and A. Meurman, "Vertex operator calculus," in: *Mathematical Aspects of String Thoery*, (San Diego, 1986), ed. by S.-T. Yau, World Scientific, Singapore, 1987, pp. 150–188.

[FLM88] I. B. Frenkel, J. Lepowsky, and A. Meurman, *Vertex Operator Algebras and the Monster*, Academic Press, New York, 1988, ISBN: 0-12-267065-5.

[Kac90] V. G. Kac, *Infinite Dimensional Lie Algebras*, 3rd ed, Cambridge University Press, Cambridge, 1990, ISBN: 0-521-46693-8.

[Kel06] J. Kelleher, "Encoding partitions as ascending compositions," PhD thesis, University College Cork, 2006.

[Lep78] J. Lepowsky, *Lectures on Kac-Moody Lie algebras*, tech. rep., Université Paris VI, 1978.

[Lep85] J. Lepowsky, "Calculus of twisted vertex operators," *Proc. Natl. Acad. Sci. USA* **82** (1985), pp. 8295–8299.

[Lep07] J. Lepowsky, "Some developments in vertex operator algebra theory, old and new," *arXiv preprint arXiv:0706.4072* (2007), URL: `http://arxiv.org/abs/0706.4072`.

[LL04] J. Lepowsky and H. Li, *Introduction to Vertex Operator Algebras and Their Representations*, Progress in Math. **227**, Birkhäuser, Boston, 2004, ISBN: 978-1-4612-6480-4.

[LM78] J. Lepowsky and S. Milne, "Lie algebraic approaches to classical partition identities," *Adv. in Math.* **29** (1978), pp. 15–59.

[LW82] J. Lepowsky and R. L. Wilson, "A Lie-theoretic interpretation and proof of the Rogers-Ramanujan identities," *Adv. in Math.* **45** (1982), pp. 21–72.

[LW84] J. Lepowsky and R. L. Wilson, "The structure of standard modules, I: Universal algebras and the Rogers-Ramanujan identities," *Invent. Math.* **77** (1984), pp. 199–290.

[LW85] J. Lepowsky and R. L. Wilson, "The structure of standard modules, II: The case $A_1^{(1)}$, principal gradation," *Invent. Math.* **79** (1985), pp. 417–442.

[MP87] A. Meurman and M. Primc, "Annihilating ideals of standard modules of $\mathfrak{sl}(2,\mathbb{C})^{\tilde{}}$ and combinatorial identities," *Adv. in Math.* **64** (1987), pp. 177–240.

[Rus13] M. C. Russell, "Noncommutative recursions and the Laurent phenomenon," *arXiv preprint arXiv:1311.1141v2* (2013), URL: `http://arxiv.org/abs/1311.1141v2`.

[Seg81] G. Segal, "Unitary representations of some infinite dimensional groups," *Commun. Math. Phys.* **80** (1981), pp. 301–342.

[TX95] M. Tamba and C. Xie, "Level three standard modules for $A_2^{(2)}$ and combinatorial identities," *J. Pure Applied Algebra* **105** (1995), pp. 53–92.

[Zei12] D. Zeilberger, *NCFPS (Maple package for non-commutative formal power series)*, 2012, URL: `http://www.math.rutgers.edu/~zeilberg/tokhniot/NCFPS`.