Constructing Graded Lie Algebras

Meighan I. Dillon

September 5, 2018

Abstract

The \( \mathbb{Z} \)-grading determined by a long simple root of an affine or finite type Lie algebra arises from an adjoint or cominuscule representation of a lower rank semi-simple complex Lie algebra. Analysis of the relationship between the grading and the representation leads to an extension of Kac’s construction of nontwisted affine Lie algebras.

Keywords \( \mathbb{Z} \)-graded Lie algebra, Kac-Moody algebra, affine Lie algebra

1 Introduction

Kac’s construction of nontwisted affine Lie algebras produces a \( \mathbb{Z} \)-graded infinite dimensional Lie algebra \( g \) from the adjoint representation of a simple complex Lie algebra. The grading is determined by a so-called special root of \( g \). We describe two related constructions that produce all \( \mathbb{Z} \)-graded finite type and affine Lie algebras— twisted as well as nontwisted— where the grading is determined by a long simple root that is not special (an \( lsn \) root.) Each construction starts with a generalized cominuscule representation of a semisimple Lie algebra.

Definition 1 If \( g = \mathfrak{l}_1 \oplus \cdots \oplus \mathfrak{l}_t \), where \( \mathfrak{l}_i \) are simple complex Lie algebras, \( V = U_1 \otimes \cdots \otimes U_t \) is a generalized cominuscule representation of \( g \) provided \( U_1 \) is irreducible over \( \mathfrak{l}_1 \) with highest weight \( n_i \Lambda_i \), \( n_i \in \mathbb{Z}_+ \), \( \Lambda_i \) the fundamental weight associated to a cominuscule simple root of \( \mathfrak{l}_1 \).

Fix \( \mathfrak{g} \), a semi-simple Lie algebra over \( \mathbb{C} \). Let \( S \) be a set of positive integers. For \( i \in S \), let \( V_i \) designate an irreducible highest weight representation of \( \mathfrak{g} \) with highest weight \( \lambda_i \). The adjoint representation of \( \mathfrak{g} \) is \( \text{adj} \mathfrak{g} \). Let \( \mathcal{T}_1 \) be the submodule of \( \Lambda^2 V_1 \) with highest weight(s) of the form \( 2\lambda_1 - \alpha \), \( \alpha \) a positive root of \( \mathfrak{g} \). \( \mathcal{T}_1^c \) is the complement of \( \mathcal{T}_1 \) in \( \Lambda^2 V_1 \). For \( i > 1 \), \( \mathcal{T}_i \) is the submodule of \( V_1 \otimes V_i \) associated to highest weights greater than or equal to those of the form \( \lambda_1 + \lambda_i - \alpha \), where \( \alpha \) is any positive root of \( \mathfrak{g} \). \( \mathcal{T}_i^c \) is the complement of \( \mathcal{T}_i \) in \( V_1 \otimes V_i \).

*Partial support was through the Faculty Development Program at Georgia Tech. Joseph Landsberg’s considerable help is especially acknowledged.
Theorem 1 The Affine Algorithm Suppose $V = V_1$ is a generalized cominuscule representation of $k^g$.

Set $j = 1$.

1. If $T^c_j$ is irreducible, let $V_{j+1} = T^c_j$. Increase $j$ by 1 and repeat this step.

2. If $T^c_j = \text{adj } k^g \oplus C$, let $V_{j+1} = \text{adj } k^g$. Increase $j$ by 1 and go to step (4).

3. Otherwise $V$ is inadmissible.

4. For $i > j$, let $V_i = \text{adj } k^g$ if $i \equiv 0 \mod j$. Otherwise take $\ell \in \{1, \ldots, j-1\}$ so that $\ell \equiv i \mod j$ and let $V_i \cong V_\ell$. For $-i < 0$, take $V_{-i} = V_i^*$. The algorithm is effective, terminating with $j \leq 6$.

If $V$ is admissible, let $g_0 = k^g \oplus C \oplus C$ and, for $i \neq 0$, let $g_i = V_i$. $g = \bigoplus_{i \in \mathbb{Z}} g_i$ is a $\mathbb{Z}$-graded affine algebra, the grading determined by an lsn root of $g$. Moreover, any lsn-graded affine Lie algebra can be constructed this way.

The next result generalizes the Minuscule Algorithm as detailed in [7].

Theorem 2 The Finite Algorithm Suppose $V = V_1$ is a generalized cominuscule representation of $k^g$.

Set $j = 1$.

1. If $T^c_j = \{0\}$, go to step (4).

2. If $T^c_j$ is irreducible, take $V_{j+1} = T^c_j$. Increase $j$ by 1 and go back to step (1).

3. Otherwise $V$ is inadmissible.

4. For $i \in \{-1, \ldots, -j\}$, take $V_{-i} = V_i^*$. For $i > j$ and $i < -j$, take $V_i = \{0\}$.

The algorithm is effective, terminating with $j \leq 6$.

If $V$ is admissible, let $g_0 = k^g \oplus C \oplus C$ and, for $i \neq 0$, let $g_i = V_i$. $g = \bigoplus_{i \in \mathbb{Z}} g_i$ is a $\mathbb{Z}$-graded finite type Lie algebra, the grading determined by an lsn root of $g$. Moreover, any lsn-graded finite type Lie algebra can be constructed this way.

Remark 1 Both algorithms relate to work of B. Kostant. Let $g_0$ be a reductive Lie algebra with module $p$. Theorem 1.50 in [6] gives necessary and sufficient conditions for determining whether a Lie bracket can be imposed on $g = g_0 + p$, consonant with the action of $g_0$ on $p$. Our work starts with semi-simple $g$ and a representation $V$ and constructs a minimal $\mathbb{Z}$-graded representation space $p = \bigoplus g_i$, so that $g_{-i} = g_i^*$, $g_1 = V$, $g_0$ contains $g$, and $g_0 + p$ is an affine or finite type Lie algebra with an lsn-grading.

The rest of the paper is dedicated to background analysis and the proofs of the theorems.
2 Terminology and Notation

A Lie algebra of finite type is a simple, finite dimensional Lie algebra over \( \mathbb{C} \). A semi-simple Lie algebra is always complex semi-simple. All Lie algebras here are affine or semi-simple. Given a Lie algebra \( \mathfrak{g} \), we fix a Cartan subalgebra \( \mathfrak{h} \). \( \Delta \) is the set of roots of \( \mathfrak{g} \), \( \Delta_+ \) the positive roots, \( \Pi \) the set of simple roots, etc. The Dynkin diagram for \( \mathfrak{g} \) is \( \mathcal{D} \). We number the simple roots of a finite type Lie algebra as in [1] and [4]. Otherwise, our notation and conventions typically follow [5]. Unless indicated to the contrary, \( \alpha_i \) is a simple root. The extra root on an extended (or affine) Dynkin diagram (cf. [1]) is always \( \alpha_0 \). The labels of \( \mathcal{D} \) are as in the Tables Aff in Chapter 4 of [5]. It is often convenient to identify elements of \( \Pi \) with the nodes of \( \mathcal{D} \).

The \( \mathbb{Z} \)-span of the roots of a Lie algebra form a lattice designated \( Q \). \( \mathbb{Z}_+ \)-linear combinations of simple roots comprise \( Q_+ \). If \( \beta \) is in \( Q_+ \), write it as a \( \mathbb{Z}_+ \)-linear combination of simple roots. The support of \( \beta \) is then the set of simple roots with positive coefficients. The height of \( \beta \) is the sum of those coefficients. When applied to roots, the words highest and lowest refer to height. Let \( \alpha_i \) be the label of \( \mathcal{D} \) associated to \( \alpha_i \). In the finite case, the highest root of \( \mathfrak{g} \) is \( \theta = \sum_{i=1}^{\ell} a_i \alpha_i \); in the affine case, the minimal positive imaginary root is \( \delta = \sum_{i=0}^{\ell} a_i \alpha_i \).

A simple root of \( \mathfrak{g} \) is special if it is conjugate to \( \alpha_0 \) under a diagram automorphism. This makes sense for a finite type \( \mathfrak{g} \) if we consider the extended Dynkin diagram for \( \mathfrak{g} \). A cominuscule root is a special root for a finite type Lie algebra (cf. [8].) The cominuscule roots are: all \( \alpha_i \) in \( A_n \); \( \alpha_1 \) in \( B_n \); \( \alpha_n \) in \( C_n \); \( \alpha_1, \alpha_{n-1} \) and \( \alpha_n \) in \( D_n \); \( \alpha_1 \) and \( \alpha_6 \) in \( E_6 \); \( \alpha_7 \) in \( E_7 \).

Let \( V \) be a representation of \( \mathfrak{g} \). A weight vector \( v^+ \) is a highest weight vector of \( V \) provided \( e_{\alpha} v^+ = 0 \) for all positive root vectors \( e_{\alpha} \) in \( \mathfrak{g} \). The weight associated to \( v^+ \) is then a highest weight. \( f_{\alpha} \) designates a root vector in \( \mathfrak{g} \) associated to \( -\alpha \), where \( \alpha \) is a positive root. A weight vector \( v^- \) is a lowest weight vector provided \( f_{\alpha} v^- = 0 \) for all positive roots \( \alpha \). To indicate that \( V \) is an irreducible \( \mathfrak{g} \)-module with highest weight \( \lambda \), we sometimes write \( V(\lambda) \) instead of \( V \).

Suppose \( \mathfrak{g} \) is finite type or affine. Distinguishing a long simple root of \( \mathfrak{g} \), \( \alpha_k \), we get a \( \mathbb{Z} \)-grading on \( \mathfrak{g} \) by \( \deg e_j = -\deg f_j = \delta_{jk} \) (Kronecker delta). (See [5], §1.5.) Here \( e_i, f_i \) are root vectors of \( \mathfrak{g} \) associated to simple \( \alpha_i \) and \( -\alpha_i \), respectively. The \( \mathbb{Z} \)-grading induced by \( \alpha_k \) is the \( \alpha_k \)-grading of \( \mathfrak{g} \). If \( \alpha_k \) is an lsn root, we say the \( \alpha_k \)-grading of \( \mathfrak{g} \) an lsn-grading.

Mark the nodes of \( \mathcal{D} \) adjacent to \( \alpha_k \). If a marked node represents a shorter root, label it with the number of edges between it and \( \alpha_k \). Excising \( \alpha_k \) and adjacent edges from \( \mathcal{D} \) we get \( \mathcal{D}_k \), a marked and labeled Dynkin diagram associated to a semi-simple Lie algebra \( \mathfrak{g}_k \) (cf. [3], Chapter 3, §3.5.) When \( \mathfrak{g} \) is affine, \( \mathfrak{g}_0 \) is the underlying algebra of finite type. \( \mathcal{D}_k \) represents an irreducible module over \( \mathfrak{g}_k \). Its highest weight is the sum of the fundamental weights associated to the marked nodes, with multiplicity according to labels. Note that in the nontwisted
affine algebras, $\mathcal{D}$ is the diagram for the adjoint representation of $\mathfrak{g}$. In general
the module associated to $\mathfrak{g}$ corresponds to $\mathfrak{g}_{-1}$, the sum of the root spaces of $\mathfrak{g}$
associated to roots with coefficient $-1$ on $\alpha_k$. The connection between the
$\alpha_k$-grading and the representation it determines is discussed thoroughly in §4
below.

Fix $\mathfrak{k} \subset \mathfrak{h}$, a Cartan subalgebra for $\mathfrak{g}$. The set of roots of $\mathfrak{g}$ is $\Delta$, the set of
simple roots is $\Pi$, etc. Fix $\Pi$ so there is a one-to-one correspondence between $\Pi$ and $\Pi - \{\alpha_k\}$ and use the same symbols to designate $\alpha_j$ in $\Pi$ and in $\Pi$, for $j \neq k$. Let $\langle . , . \rangle$ designate a fixed standard invariant form on $\mathfrak{g}$. Sometimes we
call it the Killing form. Normalize the form so that $\|\alpha\|^2 = 2$ for any long root
$\alpha$ of $\mathfrak{g}$. We use the same notation for the form as it restricts to $\mathfrak{h}$, also to $\mathfrak{k}$
and $\mathfrak{h}$. The canonical isomorphism determined by the form is $\nu : \mathfrak{h} \rightarrow \mathfrak{h}^*$ and
$\langle . , . \rangle$ designates the induced form on $\mathfrak{h}^*$ as well. Let $\langle . , . \rangle$ denote the pairing
of $\mathfrak{g}$ and $\mathfrak{g}^*$, its dual. If $\alpha \in \Delta$, $\alpha^\vee \in \mathfrak{h}$ designates its coroot. If $\alpha_i$ is long,
$\langle \alpha_i , \alpha_j^\vee \rangle$ is the number of edges between $\alpha_i$ and $\alpha_j$ in $\mathcal{D}$. For simple roots
$\alpha_i, \alpha_j, \langle \alpha_i , \alpha_j^\vee \rangle = \frac{2(\alpha_i , \alpha_j)}{\|\alpha_j\|^2}$. Also $\nu(\alpha_i^\vee) = \frac{2\alpha_i}{\|\alpha_i\|^2}$.

3 Brackets

Here we consider brackets on the $\alpha_k$-graded pieces of $\mathfrak{g} = \oplus \mathfrak{g}_i$, looking at them
in terms of the action of $\mathfrak{g}_0$ on $\mathfrak{g}_i$ by $x.g = [x, g]$. It is enough to consider $[\mathfrak{g}_1, \mathfrak{g}_j]$ and
$[\mathfrak{g}_{-1}, \mathfrak{g}_j]$ as other brackets are defined iteratively in terms of these.

If $\mathfrak{g}$ is affine type, $\mathfrak{g}_0 = \oplus \mathfrak{g}_i$, looking at them
in terms of the action of $\mathfrak{g}_0$ on $\mathfrak{g}_i$ by $x.g = [x, g]$. It is enough to consider $[\mathfrak{g}_1, \mathfrak{g}_j]$ and
$[\mathfrak{g}_{-1}, \mathfrak{g}_j]$ as other brackets are defined iteratively in terms of these.

If $\mathfrak{g}$ is affine type, $\mathfrak{g}_0 = \oplus \mathfrak{g}_i$, looking at them
in terms of the action of $\mathfrak{g}_0$ on $\mathfrak{g}_i$ by $x.g = [x, g]$. It is enough to consider $[\mathfrak{g}_1, \mathfrak{g}_j]$ and
$[\mathfrak{g}_{-1}, \mathfrak{g}_j]$ as other brackets are defined iteratively in terms of these.

3.1 $[\cdot , \cdot ] : \mathfrak{g}_{-1} \otimes \mathfrak{g}_1 \rightarrow \mathfrak{g}_0$

Let $\{X_i\}$ and $\{Y_i\}$ be Killing dual bases of $\mathfrak{g}$. If $\mathfrak{g}$ is affine, $\{X_i\} \cup \{ \frac{1}{\alpha_k} d_k , K \}$ and
$\{Y_i\} \cup \{ K , \frac{1}{\alpha_k} d_k \}$ are dual bases of $\mathfrak{g}_0$. For $u_{-1} \in \mathfrak{g}_{-1}$, $u_1 \in \mathfrak{g}_1$, we have

$$[u_{-1} u_1] = \sum_i (X_i, [u_{-1} u_1]) Y_i + \frac{1}{\alpha_k} (d_k, [u_{-1} u_1]) K + \frac{1}{\alpha_k} (K, [u_{-1} u_1]) d_k =$$

$$- \sum_i (X_i, [u_1 u_{-1}]) Y_i + \frac{1}{\alpha_k} ([d_k u_{-1}], u_1) K + \frac{1}{\alpha_k} ([K u_{-1}], u_1) d_k =$$
\[- \sum_i ((X_i u_1), u_{-1}) Y_i - \frac{1}{a_k} (u_{-1}, u_1) K.\]

Then in the affine case we have, \([u_{-1} u] = - \sum_i (u_{-1}, X_i, u_1) Y_i - \frac{1}{a_k} (u_{-1}, u_1) K.\]

If \(g\) is finite type, \(\{X_i\} \cup \{ \frac{1}{t_v} d_k \} \) and \(\{Y_i\} \cup \{d_k\}\) are dual bases of \(g_0\). For \(u_{-1} \in g_{-1}, u_1 \in g_1\), a calculation similar to the one above yields \([u_{-1} u_1] = - \sum_i (u_{-1}, X_i, u_1) Y_i - \frac{1}{t_v} (u_{-1}, u_1) d_k.\]

### 3.2 \([\cdot, \cdot] : \Lambda^2 g_1 \to g_2\)

Let \(u, v \in g_1, u_{-1} \in g_{-1}\). In the affine case, we have

\[
[u_{-1} [u v]] = [[u_{-1} u] v] + [u [u_{-1} v]] = [u_{-1} u] v - [u_{-1} v] u = - \sum_i (u_{-1}, X_i, u) Y_i, v - \frac{1}{a_k} (u_{-1}, u) K, v + \sum_i (u_{-1}, X_i, v) Y_i, u + \frac{1}{a_k} (u_{-1}, v) K, u = \sum_i (u_{-1}, X_i, v) Y_i, u - \sum_i (u_{-1}, X_i, u) Y_i, v.
\]

Viewing \([u v]\) as an element in \(\Lambda^2 g_1\), we then have

\[
[u v] = \sum_i X_i, v \wedge Y_i, u. \tag{1}
\]

A similar calculation in the finite case gives us \([u_{-1} [u v]] = - \sum_i (u_{-1}, X_i, u) Y_i, v - \frac{1}{t_v} (u_{-1}, u) d_k, v + \sum_i (u_{-1}, X_i, v) Y_i, u + \frac{1}{t_v} (u_{-1}, v) d_k, u = \sum_i (u_{-1}, X_i, v) Y_i, u - \sum_i (u_{-1}, X_i, u) Y_i, v + \frac{1}{t_v} ((u_{-1}, v) u - (u_{-1}, u) v).\]

Then in the finite case \([u v] = \sum_i X_i, v \wedge Y_i, u + \frac{1}{t_v} v \wedge u. \tag{2}\]

### 3.3 \([\cdot, \cdot] : g_{-1} \otimes g_t \to g_{t-1} \text{ and } [\cdot, \cdot] : g_1 \otimes g_t \to g_{t+1}\)

Let \(g\) be affine type and suppose \(t > 1\). Assume \([g_{-1} g_t]\) is given by

\[
[u_{-1} u_1 \ldots u_t] = - \sum_i (u_{-1}, X_i, u_1) Y_i, [u_2 \ldots u_t] - (u_{-1}, X_i, u_2) [u_1 Y_i, [u_3 \ldots u_t] - \ldots - (u_{-1}, X_i, u_{t-1}) [u_1 \ldots u_{t-2} Y_i, u_t] + (u_{-1}, X_i, u_t) [u_1 \ldots Y_i, u_{t-1}].
\]

At the same time, assume that \([g_1 g_t]\) is given by

\[
[u_1 \ldots u_t] = - \sum_i X_i, u_1 \otimes Y_i, [u_2 \ldots u_t] - X_i, u_2 \otimes [u_1 Y_i, [u_3 \ldots u_t] - \ldots - X_i, u_{t-1} \otimes [u_1 \ldots u_{t-2} Y_i, u_t] + X_i, u_t \otimes [u_1 \ldots Y_i, u_{t-1}] -
\]

5
From there it follows that

\[ u \in \{u_1 \ldots u_t \} = \{ [u_1 u_1 \ldots u_t] + [u u_{-1} u_1 \ldots u_t] = \]

\[- \sum_i (u_{-1}, X_i, u) \otimes Y_i [u_1 \ldots u_t] - \sum_i (u_{-1}, X_{i+1}, u) [u Y_i [u_1 \ldots u_t]] - \]

\[- (u_{-1}, X_i, u_i) [u u_1 \ldots u_t] + (u_{-1}, X_i, u_i) [u u_1 \ldots u_i]. \]

From there it follows that

\[ [u u_1 \ldots u_t] = - \sum_i X_i [u \otimes Y_i [u_1 u_2 \ldots u_t] - X_i [u Y_i [u_2 \ldots u_t]] - \]

\[- X_i u_{t-1} \otimes [u u_1 \ldots u_{t-2} Y_i, u_t] + X_i [u_{t-1} \otimes [u u_1 \ldots u_{t-2} Y_i, u_{t-1}]. \] (3)

Let \( \mathfrak{g} \) be finite type and suppose \( t > 1 \). Applying induction and the Jacobi identity as above to both \( [u_{-1} u_1 \ldots u_t] \) and \( [u u_1 \ldots u_t] \) we get

\[ [u u_1 \ldots u_t] = - \sum_i X_i [u \otimes Y_i [u_1 u_2 \ldots u_t] - X_i [u Y_i [u_2 \ldots u_t]] - \]

\[- X_i u_{t-1} \otimes [u u_1 \ldots u_{t-2} Y_i, u_t] + X_i [u_{t-1} \otimes [u u_1 \ldots u_{t-2} Y_i, u_{t-1}]] - \]

\[- \frac{1}{t_k} u_{t-1} \otimes [u u_1 \ldots u_t] + \frac{1}{t_k} u_t \otimes [u u_1 \ldots u_{t-1}]. \] (4)

4 The \( \alpha_k \)-Grading

This section is an elaboration on ideas sketched in [3], Chapter 3, §3.5.

The \( i \)-th \( \alpha_k \)-graded piece of \( \mathfrak{g} \), \( \mathfrak{g}_i \), is a sum of root spaces of \( \mathfrak{g} \) associated to roots \( \alpha \) with the form \( \alpha = i \alpha_k + \sum_j \beta_j \). Note that \( \mathfrak{g}_\alpha \subset \mathfrak{g}_i \) if and only if \( \mathfrak{g}_\alpha \subset \mathfrak{g}_i \).

If \( \mathfrak{g} \) is affine, suppose it is of type \( X_N^{(r)} \). Recall that \( \alpha \) is a real root of \( X_N^{(r)} \) if and only if \( \alpha + m \beta \) is a real root for all \( m \in \mathbb{Z} \). In the finite case, \( \alpha_k \) is maximal so that \( \mathfrak{g}_{\alpha_k} \) is nonzero. In the affine case, \( \mathfrak{g}_{\alpha_k} \) contains \( \mathfrak{g}_{\alpha_i} \) with some positive real root spaces. The lowest root associated to \( \mathfrak{g}_{\alpha_k} \) has the form \( r \beta - \theta \), where \( \beta \) is a highest root of \( \mathfrak{g} \). (\( \mathfrak{g} \) is semi-simple in general so there may be more than one highest root.) This is an immediate consequence of the definition of the \( \alpha_k \)-grading along with Proposition 6.3 (d) in [5]. If \( \mathfrak{g} \) is finite type, let \( \alpha_k = a_k \).

Let \( \sum_{i=0}^t c_i \alpha_i = \alpha \in \Delta_+ \). The root diagram \( D(\alpha) \) is the labeled subdiagram of \( D \) comprised of the nodes and connecting edges associated to simple roots \( \alpha \) in the support of \( \alpha \). If \( c_i > 1 \), the node associated to \( \alpha_i \) is labeled \( c_i \). We can extend the notion of a root diagram to apply to any element of \( Q_+ \). We distinguish \( \alpha_k \) in \( D(\alpha) \) by coloring the associated node. Say \( \beta \in \Delta_+ \) is a subroot of \( \alpha \in \Delta_+ \) provided \( \alpha - \beta \) is in \( Q_+ \).
Lemma 1 Let \( \gamma_i \) be a lowest root associated to \( \mathfrak{g}_i \) so that \( \gamma_1 = \alpha_k \).

\[
\gamma_2 - (\gamma_1 + \alpha_k) = \gamma_2 - 2\alpha_k = \sum_{j \neq k} c_j \alpha_j = \beta_1 + \beta_2 + \beta_3
\]

where \( \beta_i \in \Delta_+ \) have the following properties.

1. \( \beta_1 = \sum_{j \neq k} c_j' \alpha_j \) is a maximal subroot of \( \sum_{j \neq k} c_j \alpha_j \), in the sense that \( \beta_1 \) is a root, \( 0 \leq c_j' \leq c_j \) for all \( j \), and if \( c_j' \) were replaced with \( c_j' + 1 \), for any \( j \), then \( \beta_1 \) would no longer be a root, or \( c_j' + 1 \) would exceed \( c_j \). \( \beta_2 \) is a maximal subroot of \( \sum_{j \neq k} (c_j - c_j') \alpha_j \in Q_+ \) and \( \beta_3 = \sum_{j \neq k} c_j \alpha_j - \beta_1 - \beta_2 \).

2. If \( i < j \) and \( \beta_i \) and \( \beta_j \) come from the same simple component of \( \mathfrak{g} \), then \( D(\beta_j) \subseteq D(\beta_i) \).

3. \( S = \{ \beta_1, \beta_2, \beta_3 \} \) is an inert set of roots, that is, neither the sum nor the difference of elements in \( S \) is a root. In particular, \( (\beta_i, \beta_j) = 0 \) for \( i \neq j \).

4. For any arrangement of indices, \( \gamma_1 + \beta_i, \gamma_1 + \beta_i + \beta_j \) and \( \gamma_1 + \beta_i + \beta_j + \beta_k \) are all roots of \( \mathfrak{g} \).

5. \( \beta_3 \) is simple.

If \( 2 < i < ra_k \), then \( \gamma_i - (\gamma_i - 1 + \alpha_k) = \beta_1 + \beta_2 \) where \( \beta_1, \beta_2 \in \Delta_+ \). Mutatis mutandis, \( \gamma_i, \beta_i, \) and \( \beta_2 \) satisfy the first four properties above.

Proof Any element of \( Q_+ \) is a sum of successively maximal subroots as per the first statement of the lemma. We have to show that there are three such subroots for \( \gamma_2 \) and, when \( 2 < i < ra_k \), two for \( \gamma_i \). We do so by demonstrating how to construct the \( \beta_i \)s in each case. It is a routine matter to check that the other items in the lemma then follow. Constructing the \( \beta_i \)s is straightforward once we identify the possible root diagrams for \( \gamma_i \).

The tables of roots in [1] give us \( \gamma_i \) in the finite cases. In the affine cases, we appeal to Proposition 6.3 in [5]. If \( i < a_k \), \( \gamma_i \) is a finite type root, that is, its support is contained in \( \Delta \) and it appears on a list in [1]. If \( i = ja_k \), \( \gamma_i = j\delta - \theta \), where \( \theta \) is a highest long or short root of \( \mathfrak{g} \). (The root is short if \( j < r \)). The one case left to consider is when \( i > a_k \) is not a multiple of \( a_k \). That happens only if \( \mathfrak{g} = E_6^{(2)}, a_k = 2 \), and \( i = 3 \). Then \( \gamma_3 = j\delta - \xi \), where \( \xi \) is the highest short root of \( \mathfrak{g} \) with coefficient 1 on \( \alpha_k \). It turns out this is also a root of finite type. It appears as Case 5 in Figure 2.

We list all possible root diagrams for \( \gamma_2 \) in Figure 1. Along with each root diagram \( D(\gamma_2) \) is the decomposition of \( \gamma_2 - 2\alpha_k \) into a sum of three maximal subroots. These disconnected root diagrams are \( D(\beta_1), D(\beta_2), D(\beta_3) \) in each case.

Figure 2 shows the possibilities for \( D(\gamma_3) \) when \( ra_k \geq 3 \). Note that at this stage, all roots are of finite type.
Figure 1: $\gamma_n = 2\alpha_k = \beta_1 + \beta_2 + \beta_3$

Figure 2: $\gamma_3 = \gamma_2 + \alpha_k = \beta_1 + \beta_2$
Along with each root diagram is the decomposition of \( \gamma_3 \) into \( \gamma_2 + \alpha_k \) and two maximal subroots. As in Figure 1, these disconnected root diagrams are \( D(\beta_1) \) and \( D(\beta_2) \) in each case. In these and higher order cases, we have to match \( \gamma_2 \) to \( \gamma_3 \). With the catalog given in Figure 1, this is not difficult.

A complete catalog of root diagrams goes through \( a_k = 6 \). We leave the remaining three figures to the reader. \( \square \)

### 4.1 The Affine Case

When \( g \) is affine, \( \nu(K) = \delta \). Designating by \( \alpha^\vee_i \) the labels on the diagram for the Lie algebra dual to \( g \), we have \( K = \sum_{i=0}^t a_i^\vee \alpha_i^\vee \): in general \( \nu(\alpha^\vee_i) = \frac{1}{\alpha_i} \alpha_i \) (cf. [5], §6.2.) Note finally that \( h^* = h^* + C \delta + C \Lambda_k \).

Next is a record of some facts we use throughout our discussion. All are either stated explicitly in [5] or easily deduced.

#### Lemma 2

\[
\begin{align*}
\nu(\alpha_i^\vee) &= \alpha_k \\
\nu(K) &= \delta \\
\nu(d_k) &= \Lambda_k \\
(\Lambda_k, \Lambda_k) &= 0 \\
(K, K) &= 0 \\
(\delta, \delta) &= 0 \\
(\alpha_i^\vee, d_k) &= \delta_{ik} \\
(\Lambda_k, \alpha_k) &= 1 \\
(K, d_k) &= a_k \\
(\Lambda_k, \delta) &= \alpha_k^\vee \\
\Lambda_k(K) &= \alpha_k^\vee \\
<\delta, \alpha_i^\vee> &= 0 \\
<\delta, d_k> &= \alpha_k \\
<\alpha_i, d_k> &= \delta_{ik} \\
\end{align*}
\]

Let \( \pi : h^* \rightarrow h^* \) be given by \( \pi(\alpha_i) = \alpha_i, \) for \( i \neq k \), \( \pi(\delta) = 0 \), and \( \pi(\Lambda_k) = 0 \). Since \( \delta = \sum a_i \alpha_i \), this gives us \( 0 = \pi(a_k \alpha_k + \sum_{i \neq k} a_i \alpha_i) = a_k \pi(\alpha_k) + \sum_{i \neq k} a_i \alpha_i \) so that \( \pi(\alpha_k) = -\sum_{i \neq k} \frac{a_i}{\alpha_k} \alpha_i = -\frac{1}{\alpha_k} \delta + \alpha_k \).

Note that \( \alpha_k \) is special if and only if \( ra_k = 1 \).

#### Lemma 3 \( g_1 \) is an irreducible \( k \) module. When \( \alpha_k \) is special, \( g_1 \) is isomorphic to the adjoint representation of \( \mathfrak{g} \).

**Proof** \( \mathfrak{g} \) acts on \( g_1 \) via the bracket. That this defines \( g_1 \) as a \( \mathfrak{g} \) module follows from the Jacobi identity on \( \mathfrak{g} \) and the observation that \([g_0, g_1] \subset g_1 \) while \( \mathfrak{g} \subset g_0 \).

As a \( \mathfrak{g} \) module, \( g_1 \) has lowest weight \( \pi(\alpha_k) = -\frac{1}{\alpha_k} \delta + \alpha_k \). We show that up to scalar multiples, \( e_{\alpha_k} = e_k \in \mathfrak{g}_{\alpha_k} \) is the unique lowest weight vector for \( g_1 \). From there, it follows that \( g_1 \) is irreducible. Assume first that \( ra_k > 1 \).

Suppose \( \pi(\beta) \) is another lowest weight associated to \( g_1 \) so that for \( \beta \in \Delta_+ \), \( e_\beta \in \mathfrak{g}_\beta, f_\beta, e_\beta = 0 \) for all \( \alpha_j \in \Pi \). We claim there must be \( e_i \in \mathfrak{g} \) with \( e_i, e_\beta \neq 0 \).

By Lemma 2.1 in [2], we can write \( \beta = \alpha_k + \sum_{j=1}^m \alpha_{ij} \) where \( \alpha_{ij} \in \Pi \) and for all \( t \in \{1, \ldots, m\} \), \( \alpha_k + \sum_{j=1}^t \alpha_{ij} \in \Delta_+ \). We also know that if \( \beta \) and \( \beta - \alpha_j \) are
roots, then \( f_j, g_\beta \neq 0 \) (cf. [5], Prop. 3.6.) If \( f_j, e_\beta = 0 \), then \( \dim g_\beta > 1 \), so \( \beta \) is an imaginary root and \( a_k = 1 \). Since \( ra_k > 1 \), it follows that \( g \) is twisted affine. Now \( \alpha_k \) and \( -\alpha_k \) are long roots, and in the twisted affine algebras, \( \delta + \alpha \) is not a root when \( \alpha \) is long (cf. [5], Prop. 6.3.) So if \( f_i, e_\beta = 0 \) for all \( \alpha_i \in \Pi \), then \( f_k, e_\beta \) is also zero. By the same argument, if we assume \( e_i, e_\beta = 0 \) for all \( \alpha_i \in \Pi \), then \( e_k, e_\beta = 0 \) since \( \beta + \alpha_k \) is not a root. This gives us a root vector, \( e_\beta \), that commutes with all root vectors and everything in \( \mathfrak{h} \) except \( d_k \). This means \( e_\beta \) is a multiple of \( K \), which is absurd. Conclusion: if we insist that \( f_i, e_\beta = 0 \) for all \( i \in \Pi \), then (1) \( \beta \) is imaginary, and (2) there must be \( c_i, e_\beta \) with \( e_i, e_\beta \neq 0 \).

Since \( \beta + \alpha_i \) is real, the argument above gives us \( f_j \) with \( f_j, e_\alpha, e_\beta = 0 \). If \( \alpha_i = \alpha_j \), we have \( f_j, e_i, e_\beta = -\alpha_i, e_\beta + c_i, f_j, e_\beta = e_i, f_j, e_\beta = 0 \) as \( \beta, \alpha_i \neq 0 \). If \( \alpha_i \neq \alpha_j \), \( f_j \) and \( e_i \) commute and we again get \( e_i, f_j, e_\beta \neq 0 \), contradicting \( f_j, e_\beta = 0 \). Final conclusion in case \( ra_k > 1 \): up to scalar multiples, \( e_k \) is a unique lowest weight/root vector associated to \( g_1 \) so \( g_1 \) is irreducible.

Next suppose \( ra_k = 1 \) so that \( \alpha_k \) is special and \( g \) is nontwisted affine. \( k \) is then simple and \( g_\delta \subset g_1 \). The real roots of \( g_1 \) are precisely those of the form \( \alpha + \delta \), where \( \alpha \) is any root of \( \mathfrak{g} \): this follows from [5], Prop. 6.3. Thus there is a bijective correspondence between the roots of \( \mathfrak{g} \) and the real roots/weights of \( g_1 \). In particular, the highest weight of \( g_1 \) is \( \pi(\theta + \delta) = \theta \), the highest root of \( \mathfrak{g} \). It follows that there is a copy of \( \mathfrak{g} \) inside \( g_1 \). Note that all real root/weight spaces of \( g_1 \) are one dimensional and that the dimension of \( g_\delta \) is the rank of \( g \). By dimension, \( g_1 \) must be isomorphic to the adjoint representation of \( \mathfrak{g} \). (Note that \( g_\delta \) is a Cartan subalgebra.) Since \( \mathfrak{g} \) is simple, it follows that \( g_1 \) is irreducible in this case as well. □

**Lemma 4** \( g_i \) and \( g_{-i} \) are dual representations of \( g \).

**Proof** The action of \( g \) on \( g_i \) is via the bracket and it follows that \( g_i \) is a \( g \) module. As \( g_0 \) is the direct sum of \( g \) and two copies of the trivial representation of \( g \), it is self-dual. Assume \( i > 0 \). A lowest root associated to \( g_{-i} \) has the form \( -\beta = -i\alpha_k - \sum_{j \neq k} c_j \alpha_j \) where \( \sum_{j \neq k} c_j \) is maximal so that \( \beta \) is a positive root of \( g \). This is precisely the criterion that determines that \( \beta \) is a highest root associated to \( g_i \). Thus as \( g \) modules, \( g_i \) has a highest weight \( \pi(\beta) \) if and only if \( g_{-i} \) has lowest weight \( \pi(-\beta) = -\pi(\beta) \). This is to say that for \( i \neq 0 \), \( g_i \) and \( g_{-i} \) are dual \( g \) modules. □

**Lemma 5** As a \( g \) module, \( g_{ra_k} \) is isomorphic to the adjoint representation of \( g \).
Proof Since $\delta = \sum_{i=0}^{k} a_i \alpha_i$, $\mathfrak{g}_{r \delta} \subset \mathfrak{g}_{r a_k}$. All other roots $\beta = a_k \alpha_k + \sum_{j \neq k} c_j \alpha_j$ associated to $\mathfrak{g}_{r a_k}$ are real in which case $\beta - r \delta$ is a real root of $\mathfrak{g}$ and the associated root space is one dimensional. Since $\alpha_k$ is not in the support of $\beta - r \delta$, $\mathfrak{g}_{\beta - r \delta} \subset \mathfrak{g}$. Conversely, if $\mathfrak{g}_{\beta}$ is a root space in $\mathfrak{g}$, then $r \delta - \beta$ is a real root of $\mathfrak{g}_{r a_k}$. Thus root spaces in $\mathfrak{g}$ are in one-to-one correspondence with real root spaces of $\mathfrak{g}_{r a_k}$. In particular, highest weights of $\mathfrak{g}_{r a_k}$ correspond precisely to the highest roots of $\mathfrak{g}$. (In general, $\mathfrak{g}$ is semi-simple so there may be more than one highest root.) Since $\delta | \mathfrak{h} = 0$ and $\dim \mathfrak{g}_{r \delta} = \text{rank}^k \mathfrak{g}$ (cf. [5])

Corollary 8.3), $\mathfrak{g}_{r \delta} \subset \mathfrak{g}_{r a_k}$ corresponds to a Cartan subalgebra of $\mathfrak{g}$. It follows that $\mathfrak{g}_{r a_k}$ is isomorphic to the adjoint representation of $\mathfrak{g}$. □

**Lemma 6** For $i, j \neq 0, i \equiv j \mod ra_k$ if and only if, as $\mathfrak{g}$ modules, $\mathfrak{g}_i \cong \mathfrak{g}_j$.

Proof Assume $i$ and $j$ are nonzero and that $i \equiv j \mod ra_k$.

Lemma 5 and its proof remain valid if we replace $r \delta$ with $n r \delta$ and $ra_k$ with $nra_k$ for $n \in \mathbb{Z}$. In particular, $\beta \in \mathfrak{h}^*$ determines a root space in $\mathfrak{g}_{nra_k+i}$ if and only if $\beta - nr \delta$ determines a root space in $\mathfrak{g}_i$. This shows that $\mathfrak{g}_i$ and $\mathfrak{g}_{nra_k+i}$ have the same weights, thus, are isomorphic.

Next assume $\mathfrak{g}_i$ and $\mathfrak{g}_j$ are isomorphic as $\mathfrak{g}$ modules. By Lemma 4, we may assume $i$ and $j$ are both positive or both negative so say $i, j > 0$. A highest root associated to $\mathfrak{g}_j$ has the form $\beta = j \alpha_k + \sum_{m \neq k} c_m \alpha_m$ and a highest root associated to $\mathfrak{g}_i$ has the form $\gamma = i \alpha_k + \sum_{m \neq k} c'_m \alpha_m$. Since $\beta$ and $\gamma$ belong to $Q_+$, and since $\gamma - \beta$ restricted to $\mathfrak{h}$ is identically zero, it follows that $\gamma - \beta = n \delta$ for some $n \in \mathbb{Z}$. In the nontwisted algebras, this is enough to prove the result.

Now suppose $\mathfrak{g}$ is type $X_N^{(r)}$, $r = 2$ or 3. In all cases, the lowest root associated to $\mathfrak{g}_1$ is $\gamma_1 = \alpha_k$. Comparing Tables Aff 2 and 3 in [5] to Figure 1 above, we see that $\gamma_2$ is given by Case 1, 6, 7, 8, or 9. In no case is $\gamma_2 = \delta + \gamma_1$. This finishes the proof for the cases where $r = 2$ and $a_k = 1$ as $\mathfrak{g}_1 \not\cong \mathfrak{g}_2$ and for $i > 2$, $\mathfrak{g}_i$ is isomorphic to $\mathfrak{g}_1$ or $\mathfrak{g}_2$. When $\mathfrak{g} = D^{(3)}_4$, $\gamma_2$ is as in Case 7 of Figure 1. Using the numbering of roots in [5], we find

$$\gamma_3 = 2\alpha_0 + 5\alpha_1 + 3\alpha_2 = 3\delta - (\alpha_0 + \alpha_1) = 2\delta + (\alpha_1 + \alpha_2).$$

(Note that $2\delta + \alpha_k$ is not a root as $\alpha_k$ is long.) Here again, $\gamma_i + \delta$ is not a lowest root for $i = 1, 2$ or 3 so the result holds. Finally, when $\mathfrak{g} = E_6^{(2)}$ and $a_k = 2$, $\gamma_3$ is as in Case 5 of Figure 2. $\gamma_3 - \delta$ is not a root (again, $\gamma_3$ is long) and $\gamma_3 - 2\delta$ is a negative root so no root of the form $\gamma_3 + n \delta$ is a lowest positive root for $\mathfrak{g}_i$. A lowest root for $\mathfrak{g}_4$, has the form $2\delta - \theta$, $\theta$ a highest root of $\mathfrak{g}$. Since $\delta = A_3 \oplus A_1$, this confirms that no two of $\gamma_1, \gamma_2, \gamma_3$, or $\gamma_4$ differ by a multiple of $\delta$. The result follows since $\mathfrak{g}_i$ is determined by $\mathfrak{g}_i \cong \mathfrak{g}_{i+4}$ □
In any affine (or finite type) Lie algebra, \( a_k \leq 6 \). Lemmas 4, 5, and 6 together thus imply that \( g_1, g_2, \) and \( g_3 \) determine all other graded pieces of \( g \), either directly or by duality.

The next theorem gathers some of the results we established in the lemmas.

**Theorem 3**

1. The highest weight of \( g_1 \) is \( \lambda = -\frac{1}{\alpha_k} \delta - \alpha_k \). The highest weight of \( g_1 \) is \( \frac{1}{\alpha_k} \delta - \sigma(\alpha_k) \), where \( \alpha_k \) and \( \sigma(\alpha_k) \) are dual roots, that is, conjugate under an automorphism of the Dynkin diagram.

2. The highest weight of \( g_1 \) is also given by \( \lambda = -\frac{1}{\alpha_k} \delta - \sigma(\alpha_k) \), where \( \alpha_k \) and \( \sigma(\alpha_k) \) are dual roots, that is, conjugate under an automorphism of the Dynkin diagram.

3. The grading on an affine Lie algebra \( X_{\mathbb{N}}(r) \) determined by a long simple root \( \alpha_k \) has period \( ra_k \).

4. The grading defined by a long simple root \( \alpha_k \) on an affine Lie algebra is completely determined by \( g_k, g_1, g_2 \) and \( g_3 \).

**Proof** It is clear that \( \lambda = \pi(-\alpha_k) = -\frac{1}{\alpha_k} \delta - \alpha_k \) is the highest weight of \( g_1 \). The second statement follows from the fact that \( g_1 \) and \( g_1 \) are dual representations.

If \( \alpha_i \) is a simple root of \( k \), \( < \lambda, \alpha_i^\vee > = < -\alpha_k, \alpha_i^\vee > \) is the number of edges between \( \alpha_k \) and \( \alpha_i \) in \( D \), since \( \alpha_k \) is long. The form of \( \lambda_1 \) follows by duality. Since there is one component of \( D \) for each simple root adjacent to \( \alpha_k \) in \( D \), \( g_1 \) is generalized cominuscule representation of \( k \).

The rest of the results have already been established. \( \square \)

**4.2 The Finite Case**

Let \( g \) be finite type so that \( \mathfrak{h} = \mathfrak{h}^k \oplus \mathbb{C}d_k \) and \( \mathfrak{h}^* = \mathfrak{h}^* \oplus \mathbb{C}\Lambda_k \) with \( d_k = \nu^{-1}(\Lambda_k) \).

It is clear that with respect to the \( \alpha_k \)-grading on \( g \), \( g_0 = g \oplus \mathbb{C}d_k \). The following is an analog to Lemma 2.

**Lemma 7**

\[
(\alpha_i^\vee, d_k) = \delta_{ik} \quad (\Lambda_k, \alpha_k) = \frac{\alpha_i^\vee}{\alpha_k} = 1 \quad \nu(\alpha_i^\vee) = \alpha_k
\]

\[
\nu(d_k) = \Lambda_k \quad < \alpha_i, d_k > = \delta_{ik} \quad \square
\]

Project \( \mathfrak{h}^* \) onto \( \mathfrak{h}^* \) by \( \pi : \mathfrak{h}^* \rightarrow \mathfrak{h}^* \) where \( \pi(\alpha_i) = \alpha_i \), for \( i \neq k \), and \( \pi(\Lambda_k) = 0 \).

If \( \Lambda_k = \sum_j f_j \alpha_j \) we have \( \pi(\alpha_k) = -\sum_{i \neq k} \frac{f_i}{f_k} \alpha_i = -\frac{1}{f_k} \Lambda_k + \alpha_k \).
Lemma 8 \( g_1 \) is an irreducible \( \mathfrak{g} \) module.

Proof The first part of the proof of Lemma 3 applies here. We do not need a separate argument for the case \( a_k = 1 \) because in the current setting, all roots of \( \mathfrak{g} \) are real. □

The proof of Lemma 4 applies nearly unchanged to give us the following.

Lemma 9 \( g_i \) and \( g_{-i} \) are dual representations of \( \mathfrak{g} \). □

Let \( \theta \) be the highest root of \( \mathfrak{g} \) so that \( \theta = \sum_{i>0} a_i \alpha_i \). Let \( \Lambda = \pi(\theta) \) so

\[ \Lambda = \sum_{i \neq k} \left( a_i - \frac{t_i}{t_k} \right) \alpha_i. \]

Proposition 1 As \( \mathfrak{k} \) modules, \( g_{a_k} \cong V(\Lambda) \).

Proof \( g_{a_k} \) is a sum of root spaces, among them \( g_\theta \). Since \( \theta \) is the highest root associated to \( \mathfrak{g} \), it is the highest root associated to \( g_{a_k} \). The highest weight associated to \( g_{a_k} \) as a \( \mathfrak{g} \) module is thus \( \Lambda = \pi(\theta) \). All roots here are real so \( \dim g_\theta = 1 \). If \( v^+ \) is a highest weight vector associated to another root space in \( g_{a_k}, g_\beta \), then \( \beta \) has the property that \( \beta + \alpha_i \) is not a root for any \( i \neq k \). Since \( a_k \) is the coefficient of \( \alpha_k \) in \( \theta \) as well as in \( \beta \), we cannot add \( a_k \) to \( \beta \) and get a root of \( \mathfrak{g} \) either. Thus, \( \beta + \alpha_i \) is a root for any simple root \( \alpha_i \) associated to \( \mathfrak{g} \), which is impossible unless \( \beta = \theta \), thus, \( g_{a_k} \) is irreducible as a \( \mathfrak{g} \) module. □

Remark 2 If \( a_k > 1 \), \( \dim g_{a_k-1} = \frac{1}{a_k-1} \dim g_1 \). The reader can verify this curious fact case by case for \( \mathfrak{g} \) finite type but we have no insight beyond the observation itself. There appears to be no analog for affine algebras.

Theorem 4 1. The highest weight of \( \mathfrak{g}_{-1} \) is \( \lambda_{-1} = \frac{1}{t_k} \Lambda_k - \alpha_k \), where \( \Lambda_k = \sum_{i \neq k} t_i \alpha_i \). The highest weight of \( \mathfrak{g}_1 \) is \( \lambda_1 = -\frac{1}{t_k} \Lambda'_k + \sigma(\alpha_k) \) where \( \Lambda'_k \) is fundamental on \( \sigma(\alpha_k) \), the root dual to \( \alpha_k \), and \( \Lambda'_k = \sum_{i} t'_i \alpha_i \).

2. The highest weight of \( \mathfrak{g}_{-1} \) is also given by \( \lambda_{-1} = \sum_{i \neq k} s_i \Lambda_i \) where \( s_i \) is the number of edges in \( D \) connecting \( \alpha_i \) and \( \alpha_k \). \( g_1 \) has highest weight \( \lambda_1 = \sum_{i \neq k} s'_i \Lambda_i \) where \( s'_i \) is the number of edges in \( D \) connecting \( \alpha_i \) and \( \sigma(\alpha_k) \). In particular, \( g_1 \) is generalized cominuscule.

Proof It is clear that the highest weight associated to \( \mathfrak{g}_{-1} \) is \( \pi(-\alpha_k) = \frac{1}{t_k} \Lambda_k - \alpha_k \). For \( i \neq k \), \( \lambda_{-1}, \alpha_i^\vee > \lambda_{-1}, \alpha_i^\vee \), which is the number of edges between \( \alpha_k \) and \( \alpha_i \) in \( D \), since \( \alpha_k \) is long. The forms of \( \lambda_1 \) follow by duality. As in the affine case, \( \mathfrak{g} \) has one simple component per simple root adjacent to \( \alpha_k \) in \( D \). This, along with the form of \( \lambda_1 \) given in the second statement, gives us that \( g_1 \) is generalized cominuscule. □
4.3 Representations

We revisit Lemma 1 as a statement about weights.

**Proposition 2** Let $\lambda_i$ be a highest weight of $\mathfrak{g}_i$ as a $k\mathfrak{g}_i$-module. If $a_k > 1$, then

$$\lambda_2 = 2\lambda_1 - (\beta_1 + \beta_2 + \beta_3)$$

for positive roots $\beta_i$ with properties 1-5 in Lemma 1. If $a_k \geq t > 2$,

$$\lambda_t = t\lambda_1 - \sum_{i=1}^{2t-1} \beta_i,$$

for positive roots $\beta_i$ arising as in Lemma 1. In particular, $\lambda_t = \lambda_t - 1 + \lambda_1 - \beta_1 - \beta_2$, where $\beta_1, \beta_2 \in \Delta^k_+$ satisfy properties 1-4 in Lemma 1, mutatis mutandis.

**Proof** If $\gamma_t$ is a lowest root associated to $\mathfrak{g}_t$, then $\pi(\gamma_t)$ is a lowest weight of $\mathfrak{g}_t$. Let $w$ be the longest element in the Weyl group of $k\mathfrak{g}_t$; then $w(\pi(\gamma_t)) = \lambda_t$, the highest weight of $\mathfrak{g}_t$.

Lemma 1 gives us $\gamma_2 = 2\gamma_1 + \beta_1 + \beta_2 + \beta_3$. Then

$$w(\pi(\gamma_2)) = \lambda_2 = 2w(\pi(\gamma_1 + \beta_1 + \beta_2 + \beta_3)) = 2\lambda_1 - (\beta'_1 + \beta'_2 + \beta'_3)$$

where the $\beta'_i$s are positive roots. The longest element in the Weyl group of $k\mathfrak{g}$ respects properties 1-5 of Lemma 1. Thus, the $\beta'_i$s enjoy those properties as well. The argument for $t > 2$ is similar. □

We have established that $\mathfrak{g}_i$ and $\mathfrak{g}_{-1}$ are irreducible $k\mathfrak{g}$ modules. Next we show that all $\mathfrak{g}_i$ are irreducible when $i \not\equiv 0 \mod ra_k$, in the affine case, $i \not\equiv 0$ in the finite case. First we need a technical lemma.

If $\alpha$ is a subroot of $\beta$, i.e., $\beta - \alpha \in \mathbb{Q}_+$, we write $\alpha \subseteq \beta$. When simple $\alpha_i$ is in the support of $\alpha$, we write $\alpha_i \in \alpha$.

**Lemma 10** If $\alpha$ is a positive root of $\mathfrak{g}$ and $\alpha_r \subseteq \alpha$, we can write $\alpha = \alpha_r + \sum_{i=1}^{m} \alpha_i$, for $\alpha_i \in \Pi$ where, for all $t \in \{1, \ldots, m\}$, $\alpha_r + \sum_{j=1}^{t} \alpha_i \in \Delta_+.$

**Proof** The case where $\alpha_r$ is simple is done in [2], Lemma 2.1, so here we assume that the height of $\alpha_r$ is greater than one.

If there is a simple root $\alpha_i \not\in \alpha_r$ with $\beta = \alpha - \alpha_i \in \Delta_+$, we get the result by induction on the height of $\alpha$ because $\alpha_r \not\subseteq \beta$.

If no such simple root exists, we proceed by induction on $ht(\alpha - \alpha_r)$, the case $ht(\alpha - \alpha_r) = 1$ being clear. If $ht(\alpha - \alpha_r) > 1$, the result follows once we produce a simple root $\alpha_j \in \alpha - \alpha_r$ with $\alpha_r + \alpha_j \in \Delta_+$.

If there is a simple root $\alpha_j \in \alpha$, $\alpha_j \not\in \alpha_r$, we claim there must be one that satisfies $\alpha_r + \alpha_j \in \Delta_+$. The support of any root must form a connected component of the Dynkin diagram, which implies there must be $\alpha_j$ that shares
an edge of \( D \) with some \( \alpha_i \in \alpha_r \). Then \( \langle \alpha_r, \alpha_i \rangle < 0 \) implies \( \alpha_r + \alpha_j \in \Delta_+ \) as claimed. That leaves us with the case where \( \supp \alpha = \supp \alpha_r \).

If \( \langle \alpha_r, \alpha \rangle < 0 \), there must be a simple \( \alpha_i \in \alpha \) with \( \alpha_r + \alpha_i \in \Delta_+ \). If \( \langle \alpha_r, \alpha \rangle = 0 \), invoke the fact that \( \supp \alpha_k = \supp \alpha \). This, along with the fact that \( \text{ht} \alpha_r > 1 \) implies there is \( \alpha_i \in \supp \alpha_r \) with \( \alpha_r - \alpha_i \in \Delta_+ \), means that there must be \( \alpha_j \in \supp \alpha_r = \supp \alpha \) with \( \alpha_r + \alpha_j \in \Delta_+ \). Finally, suppose \( \langle \alpha_r, \alpha \rangle > 0 \). This gives us \( \alpha - \alpha_r \in \Delta_+ \). Now invoke \( \alpha_r + (\alpha - \alpha_r) \in \Delta_+ \) to get some \( \alpha_i \in \supp (\alpha - \alpha_r) \) with \( \alpha_r + \alpha_i \) a root. \( \square \)

**Theorem 5** If \( \mathfrak{g} = X_N^{(r)} \) is affine and \( i \not\equiv 0 \bmod r a_k \), \( g_i \) is an irreducible \( \mathfrak{g} \) module. If \( \mathfrak{g} \) is finite type and \( i \not\equiv 0 \), \( g_i \) is an irreducible \( \mathfrak{g} \) module.

**Proof** To show \( g_i \) is irreducible, we show it has a unique lowest weight.

We show below that the lowest root for \( g_i \) is well-defined. For now, assume this is so and let \( \gamma_i \) be the lowest root in \( g_i \). If there is a second root \( \gamma_i' \) such that \( f_j, u = 0 \) for some nonzero \( u \) in \( g_i \), and all \( \alpha_j \in H \), note that \( \gamma_i \subseteq \gamma_i' \).

Lemma 10 allows us to write \( \gamma_i' = \gamma_i + \sum_{j=1}^m \alpha_i j \) where \( \alpha_i j \in H \) and for all \( t \in \{1, \ldots, m\} \), \( \gamma_i + \sum_{j=1}^t \alpha_i j \in \Delta_+ \). Applying Proposition 3.6 in [5], we get \( f_{it}, g_{\gamma_i'} \neq 0 \). Since \( \gamma_i \) and \( \gamma_i' \) have the same coefficient on \( \alpha_k \), \( \alpha_i j \) must belong to \( H \). Our assumption that \( f_{it}, u = 0 \) forces us to conclude that \( \dim g_{\gamma_i'} > 1 \), thus, that \( \gamma_i' \) is imaginary. An argument similar to the one we used for Lemma 6 applies to show that there must be some \( f_{it} \) with \( \alpha_i j \in H \) and \( f_{it}, u \neq 0 \). The contradiction proves that \( g_i \) is irreducible.

Now we show that there is a unique lowest root \( \gamma_i \) associated to \( g_i \).

There is only one choice of \( \gamma_i = \alpha_k \in h^* \) so the result holds when \( i = 1 \) thus when \( i = -1 \). By duality and the fact that \( a_k \leq 6 \), it suffices to show that if \( i = 2 \) or \( 3 \) and \( r a_k > 2 \) or \( 3 \) respectively, then for a fixed \( \mathfrak{g} \) and fixed \( a_k \), there is only one choice of \( \gamma_i \) from among the root diagrams in Figures 1 and 2 respectively, at least in the affine case. We deal with those details and leave the rest of the finite type cases to the reader.

A choice of \( \gamma_i \) corresponds to a choice of injective mapping of \( D(\gamma_i) \), up to automorphism of \( D(\gamma_i) \), into \( D \) with its \( a_k \) node marked. We do our accounting by considering Tables Aff 1, 2, and 3 in [5] that have nodes associated to long roots with labels such that \( r a_k > 2 \). We compare those to the diagrams from Figures 1 and 2 that inject into \( D \) and see that in each case, as long as \( i = r a_k \), there is a unique injection of \( D(\gamma_i) \) into \( D \), up to automorphism of \( D(\gamma_i) \).

Consider the case \( i = 2 \) where \( r a_k > 2 \). \( \mathfrak{g} \) can be type \( E \), \( F \), or \( D_4^{(3)} \). Since \( D_4^{(3)} \), \( E_6^{(2)} \), and \( F \) type algebras are not simply laced, it is clear that there is only one way to inject \( D(\gamma_i) \) into \( D \) in these cases. (These correspond respectively to Cases 7, 6, and 1 in Figure 1.)

Next suppose \( D \) is non-twisted type \( E \) with the branch node marked. This is Case 2 in Figure 1. There are three different ways to inject that diagram into
with the branch node distinguished but these different injections arise from diagram automorphisms of \( \mathcal{D}(\gamma_2) \).

If \( \alpha_k \) corresponds to the terminal node at the end of the shortest branch of \( \mathcal{D} \), \( \mathcal{D}(\gamma_2) \) is Case 3 in Figure 1. Ignoring diagram automorphisms, there is only one way to inject this into \( \mathcal{D} \). If \( \alpha_k \) is the terminal node at the end of a long branch, then \( g = E_8^{(1)} \) and the diagram from Figure 1 is Case 5. There is a unique injection of \( \mathcal{D}(\gamma_2) \) into \( \mathcal{D} \).

If \( g \) is non-twisted \( E \) type with some other distinguished node, the relevant diagram in Figure 1 is Case 2. If \( \mathcal{D}(\gamma_2) \) is a \( D_5 \) diagram, it fits two different ways into \( E_7^{(1)} \) and \( E_8^{(1)} \) diagrams. But with a non branch node distinguished on each diagram, there is only one way to inject \( \mathcal{D}(\gamma_2) \) into \( \mathcal{D} \).

This is a complete proof that \( g_2 \) is irreducible in case \( ra_k > 2 \). Next consider \( g_3 \) when \( ra_k > 3 \).

Here \( \mathcal{D} \) can be type \( E_6^{(2)} \), \( E_7^{(1)} \) or \( E_8^{(1)} \). Again, since \( E_6^{(2)} \) is not simply laced, the uniqueness of the diagram injection is immediate. (\( \mathcal{D}(\gamma_3) \) is Case 5 in Figure 2.) In the \( E_7^{(1)} \) case the distinguished node is the branch node and \( \mathcal{D}(\gamma_3) \) comes from Case 1 in Figure 2. Note that \( \mathcal{D}(\gamma_3) \) is a type \( E_6 \) diagram. There are two ways to inject such a diagram into one of type \( E_7^{(1)} \) but the different ways arise from an automorphism of the \( E_6 \) diagram.

When the distinguished node on \( \mathcal{D} \) is not the branch node, it is one or two nodes away from the branch node. If one away, it is adjacent to a terminal node or not. If \( \alpha_k \) is between a terminal node and a branch node, \( \mathcal{D}(\gamma_3) \) comes from Case 2 in Figure 2 so is of type \( E_7 \). There is a unique way to inject an \( E_7 \) diagram into an \( E_8^{(1)} \). If \( \alpha_k \) is adjacent to the branch, not adjacent to a terminal node, \( \mathcal{D}(\gamma_3) \) is again from Case 1 Figure 2. This also injects uniquely into \( \mathcal{D} \). \( \square \)

4.4 The Invariant Form and The Casimir Operator

How is \( (\ldots, \ldots)_{\mathfrak{k}_{\mathfrak{g}}} \) related to the Killing form on \( \mathfrak{k}_{\mathfrak{g}} \)?

If \( V(\lambda) \) is a generalized cominuscule representation of a semi-simple Lie algebra \( \mathfrak{g} \) with \( \lambda = \sum n_i \Lambda_i \) as in Definition 1, the degree of the representation on \( \mathfrak{t}_i \) is \( n_i \).

Let \( \mathfrak{t}_i \) be a simple component of \( \mathfrak{g}_{\mathfrak{k}} \). Let \( n_i \) be the degree of \( \mathfrak{g}_{\mathfrak{k}} \) as a generalized cominuscule representation of \( \mathfrak{t}_i \). If \( B(\ldots, \ldots) \) is the standard invariant form on \( \mathfrak{k} \) normalized so that the square length of a long root is 2, then

\[
(\ldots, \ldots)|_{\mathfrak{t}_i} = \frac{1}{n_i} B(\ldots, \ldots).
\]

This maintains relative root lengths in \( \mathfrak{g} \) on restriction to \( \mathfrak{k}_{\mathfrak{g}} \). Conversely, if we start with semi-simple \( \mathfrak{k} \) in a generalized cominuscule representation, and construct \( \mathfrak{g} \) using one of the algorithms, \((\ldots, \ldots)|_{\mathfrak{t}_i} := \frac{1}{n_i} B(\ldots, \ldots) \) lifts to the invariant
form on $g$ that measures the square length of a long root as 2. Note: (1) long roots of $g$ belonging to different simple components may have different lengths; (2) insisting that the square length of long roots of $g$ is 2 is not the standard normalization on twisted affine Lie algebras; (3) it remains true that $\langle \alpha, \alpha^\vee \rangle = \frac{2(\alpha, \alpha)}{\|\alpha\|^2}$ is the number of edges shared by $\alpha, \alpha^\vee \in \Pi$ and when $\alpha$ is long.

Let $\lambda_1$ be a highest weight of $g_j$.

**Lemma 11**

1. In the affine case, $(\lambda_1, \lambda_1) = 2$. In the finite case, $(\lambda_1, \lambda_1) = 2 - \frac{1}{t_k}$, where $\Lambda_k = \sum_i t_i \alpha_i$.

2. For $\alpha_i \in k $, $(\lambda_1, \alpha_i) = 1$ or 0.

**Proof**

1. In the affine case, $(\lambda_1, \lambda_1) = \|\frac{1}{\alpha_k} - \alpha_k\|^2 = \|\alpha_k\|^2 = 2$. The proof in the finite case is similar.

2. Suppose $\langle \lambda_1, \alpha_i^\vee \rangle = n$ so that $\|\alpha_i\|^2 = 2/n$. Then $(\lambda_1, \alpha_i) = \langle \lambda_1, \alpha_i^\vee \rangle = \frac{\|\alpha_i\|^2}{2} = 1$. □

If $V(\lambda)$ is an irreducible highest weight representation of a semi-simple Lie algebra $g$, the Casimir operator $C$ takes the scalar value $c = (\lambda, \lambda) + 2(\rho, \lambda)$, where $\rho$ is the sum of fundamental weights associated to $g$.

In what follows, $c_i$ is the scalar value of the Casimir operator on $g_i$ when $g_i$ is irreducible and $\rho$ is the sum of the fundamental weights associated to $g_i$.

**Proposition 3** $T_1$ is a Casimir eigenspace. In the affine case, $C|_{T_1} = 2c_1$. In the finite case, $C|_{T_1} = 2c_1 - \frac{2}{t_k}$.

**Proof** A highest weight of $T_1$ has the form $2\lambda_1 - \alpha_i$ where $\alpha_i$ is a simple root of $g$ with $\langle \lambda_1, \alpha^\vee_i \rangle \neq 0$. On the associated irreducible component of $T_1$, we have

$$C = (2\lambda_1 - \alpha_i, 2\lambda_1 - \alpha_i) + 2(\rho, 2\lambda_1 - \alpha_i) = 2c_1 + 2(\lambda_1, \lambda_1) - 4(\lambda_1, \alpha_i) + \|\alpha_i\|^2 - 2(\rho, \alpha_i).$$

In the affine case, $(\lambda_1, \lambda_1) = 2$ by Lemma 11. We also have

$$(\rho, \alpha_i) = \langle \rho, \alpha_i^\vee \rangle = \frac{\|\alpha_i\|^2}{2} = \frac{\|\alpha_i\|^2}{2}.$$ 

Since $(\lambda_1, \alpha_i) = 1$, we get

$$C = 2c_1 + 4 - 4 = 2c_1.$$

When we do the calculation for the finite case, the only change is $\|\lambda_1\|^2 = 2 - \frac{1}{t_k}$. □
Let $\beta_i$ be as in Lemma 1 and let $n$ be the degree of the representation $\mathfrak{g}_2$ on a given component of $\hat{k}$. Perusal of the diagrams in Figure 1 reveals two important data: (1) the $\beta_i$s are all long roots in their respective components of $\hat{k}$; and (2) in Cases 7-9, those for which $n > 1$, the $\beta_i$s belong to a single component of $\hat{k}$. In particular, $\|\beta_i\|^2 = 2/n$, where $n$ is well-defined in each case.

**Corollary 1**

1. Suppose $ra_k > 2$. In the affine case, the Casimir operator acts on $\mathfrak{g}_2$ as the scalar $c_2 = 2c_1 - 8 + \frac{6}{r} - \frac{2}{n} \sum_{i=1}^{3} \text{ht} \beta_i$.

2. Suppose $ra_k \geq 2$. In the finite case, the Casimir operator acts on $\mathfrak{g}_2$ as the scalar $c_2 = 2c_1 - 8 + \frac{6}{r} - \frac{2}{n} \sum_{i=1}^{3} \text{ht} \beta_i$.

**Proof**

Consider that

$$c_2 = 2(\lambda_1 - (\beta_1 + \beta_2 + \beta_3), 2\lambda_1 - (\beta_1 + \beta_2 + \beta_3)) + 2(\rho, 2\lambda_1 - (\beta_1 + \beta_2 + \beta_3))$$

$$= 4(\lambda_1, \lambda_1) + 4(\rho, \lambda_1) - 4(\lambda_1, \beta_1 + \beta_2 + \beta_3) + \|\beta_1 + \beta_2 + \beta_3\|^2 - 2(\rho, \beta_1 + \beta_2 + \beta_3)$$

$$= 2c_1 + 2(\lambda_1, \lambda_1) - 4 \cdot 3 + \|\beta_1 + \beta_2 + \beta_3\|^2 - 2(\rho, \beta_1 + \beta_2 + \beta_3).$$

We note

$$(\rho, \beta_i) = <\rho, \beta_i' > \frac{\|\beta_i\|^2}{2} = \frac{\text{ht} \beta_i}{n}$$

giving us

$$c_2 = 2c_1 + 2(\lambda_1, \lambda_1) - 12 + \frac{6}{n} - \frac{2}{n} \sum_{i=1}^{3} \text{ht} \beta_i.$$ 

Suppose $ra_k > 2$ in the affine case. Perusal of the tables in [5] reveals that $n = r$ in these cases. By Lemma 11 we have

$$c_2 = 2c_1 - 8 + \frac{6}{r} - \frac{2}{r} \sum_{i=1}^{3} \text{ht} \beta_i.$$ 

If $a_k \geq 2$ in the finite case, we apply Lemma 11 to get

$$c_2 = 2c_1 + 4 - \frac{2}{t_k} - 12 + \frac{6}{n} - \frac{2}{n} \sum_{i=1}^{3} \text{ht} \beta_i$$

$$= 2c_1 - 8 - \frac{2}{t_k} + \frac{6}{n} - \frac{2}{n} \sum_{i=1}^{3} \text{ht} \beta_i. \Box$$

Let $Y_{1,t}$ be the Cartan product of $\mathfrak{g}_1$ and $\mathfrak{g}_t$, that is, $Y_{1,t}$ is irreducible with highest weight $\lambda_1 + \lambda_t$. A corollary of Proposition 2 follows.

**Corollary 2**

1. If $\mathfrak{g}$ is affine, $(\lambda_1, \lambda_t) = 1$, for $t \in \{2, \ldots, a_k\}$. If $\mathfrak{g}$ is finite type, $(\lambda_1, \lambda_t) = 1 - \frac{1}{t_k}$. 

18
2. If \( g \) is affine, the value of the Casimir operator on \( Y_{1, t} \) is \( c_1 + c_t + 2 \). If \( g \) is finite type, the value of the Casimir on \( Y_{1, t} \) is \( c_1 + c_t + 2 - \frac{2}{k} \).

3. If \( g \) is affine, the value of the Casimir operator on a component of \( T_t \) with highest weight \( \lambda_1 + \lambda_t - \alpha \) is \( c_1 + c_t + 2 - 2(\rho, \alpha) \). If \( g \) is finite type, the Casimir on such a component is \( c_1 + c_t + 2 - \frac{2}{k} - 2(\rho, \alpha) \).

Proof

1. Proposition 2 along with Lemma 11 gives us

\[
(\lambda_1, \lambda_t) = t(\lambda_1, \lambda_1) - (2t - 1) = 2t - 2t + 1 = 1
\]

when \( g \) is affine and when \( g \) is finite type, we adjust according to \((\lambda_1, \lambda_1) = 2 - \frac{1}{k}\).

2. The Casimir operator takes the following value on \( Y_{1, t} \):

\[
(\lambda_1 + \lambda_t, \lambda_1 + \lambda_t) + 2(\lambda_1 + \lambda_t, \rho) = c_1 + c_t + 2(\lambda_1, \lambda_t).
\]

Apply statement (1) now to get statement (2).

3. If \( g \) is affine type, the Casimir on a given component of \( T_t \) is

\[
c_1 + c_t + 2 - 2(\alpha, \lambda_1 + \lambda_t) + (\alpha, \alpha) - 2(\rho, \alpha).
\]

We have \((\lambda_1, \alpha) = (\lambda_t, \alpha) = 1\) so the Casimir is \( c_1 + c_t - 2 - 2(\rho, \alpha) \). In the finite case, adjust the calculation to reflect the result in part (2). \(\Box\)

\(g_2\) is a distinguished submodule of \( \Lambda^2 g_1 \) and for \( i > 2 \), \( g_i \) is a distinguished submodule of \( g_1 \otimes g_{i-1} \). The bracket is then a projection from \( g_1 \otimes g_{i-1} \) onto \( g_i \). Proposition 2 tells us what the kernel of the projection is.

Corollary 3 The bracket as defined on \( \Lambda^2 g_1 \) is identically zero on \( T_1 \). If \( 2 \leq i \leq a_k - 1 \), the bracket defined on \( g_1 \otimes g_i \) is identically zero on \( T_i \). \(\Box\)

5 Proof of the Algorithm, Part I

It is convenient at this point to treat the Affine Algorithm and the Finite Algorithm as one process. Our goals in this section are: (1) to show that the algorithms are effective, either aborting or terminating successfully with \( j \leq 6 \); and (2) to establish that there is a one-to-one correspondence between the graded \( k \) modules produced by the algorithms and lsn-graded affine/finite type Lie algebras. (Note that \( A_{a_n}^{(1)} \) and \( C_{a_n}^{(1)} \) are the only affine or finite type Lie algebras without lsn-gradings.) In the next section, we establish that if \( g \) is produced by one of the algorithms, then it has the expected Lie algebra structure.

The effect of a successful run of either algorithm is to append a long node to \( k \). When \( V \) is degree \( n \) on a component of \( \hat{g} \), the associated node of \( \hat{D} \) shares \( n \)
edges with the appended node. We use this idea to catalog the graded modules produced by the algorithm.

When considering different input modules, we make no distinction between duals or between different modules that yield dual outputs, for example, the two half-spin representations of $D_n$.

**Lemma 12** The algorithm aborts with $j = 1$ in cases where $f$ has more than four components. The only case in which it tolerates four components is that for which $V = U_1 \otimes \ldots \otimes U_4$, where $U_i$ is the standard (two dimensional) representation of $A_1$. This case terminates with $j = 2$ and corresponds to $\alpha_3$-graded $D_4^{(1)}$.

**Proof** Suppose $f = t_1 \oplus \ldots \oplus t_5$ and $V = U_1 \otimes \ldots \otimes U_5$, $U_i$ generalized cominuscule on $t_i$. $\Lambda^2 V$ has sixteen (not necessarily irreducible) components, each a product of one, three, or five $\Lambda^2 U_i$s, with, respectively, four, two, or zero $S^2 U_j$s. A weight argument confirms that $\Lambda^2 U_i$ and four $S^2 U_j$ factors. This leaves at least eleven more irreducible components, too many to comprise either an irreducible $T_1^c$ or the adjoint representation of $f$. If $f$ has more than five components, the number of components of $\Lambda^2 V$ is even farther beyond the maximum tolerated by the algorithm. So in all cases where $f$ has five or more simple components, the algorithm aborts with $j = 1$.

Next suppose $f$ has four components. This time, $\Lambda^2 V$ has eight not necessarily irreducible components; four each consisting of the tensor product of one $\Lambda^2 U_i$ with three $S^2 U_j$s (this sum contains $T_1$), and four each consisting of the tensor product of three $\Lambda^2 U_i$s with one $S^2 U_j$ (these are contained in $T_1^c$.) Since the sum of four submodules contained in $T_1^c$ is not irreducible, the algorithm aborts with $j = 1$ unless $T_1^c$ coincides with the adjoint representation of $f$. Note in particular that the algorithm aborts if any of the eight components of $\Lambda^2 V$ is not irreducible. If the algorithm does not abort, the submodule of $\Lambda^2 V$ consisting of products of one symmetric square and three exterior squares must be of the form $\text{adj} t_i \otimes \mathbb{C} \otimes \mathbb{C} \otimes \mathbb{C}$, up to ordering of the factors. This establishes that $\Lambda^2 U_i = \mathbb{C}$, thus, that $U_i$ is the two dimensional representation of $sl_2(\mathbb{C})$. The algorithm terminates successfully then with $j = 2$. □

### 5.1 $V = U_1 \otimes U_2 \otimes U_3$

Suppose $V = U_1 \otimes U_2 \otimes U_3$, where $U_i$ is a generalized cominuscule representation of $t_i$. For the remainder of this section, we use $n_i$ to designate the rank of $t_i$. To make accounting easier, assume $n_1 \leq n_2 \leq n_3$. We have

$$\Lambda^2 V = \Lambda^2 U_1 \otimes S^2 U_2 \otimes S^2 U_3 \oplus \Lambda^2 U_2 \otimes S^2 U_3 \otimes S^2 U_2 \otimes \Lambda^2 U_1 \oplus \Lambda^2 U_1 \otimes \Lambda^2 U_2 \otimes \Lambda^2 U_3.$$
$T_i^c$ always contains $Λ^2U_1 ⊗ Λ^2U_2 ⊗ Λ^2U_3$. If some $S^2U_i$ or $Λ^2U_i$ is not irreducible, $T_i^c$ has other components as well. $S^2U_i$ is irreducible only if $U_i$ is the standard representation of $A_n$. In this case, $Λ^2U_i$ is irreducible, as well. Thus, if $T_i^c$ has components other than $Λ^2U_1 ⊗ Λ^2U_2 ⊗ Λ^2U_3$, it is because $S^2U_i$ is not irreducible.

If $S^2U_i$ is not irreducible and the algorithm advances beyond $j = 1$, then $Λ^2U_1 ⊗ Λ^2U_2 ⊗ Λ^2U_3 ∼= adj l_i$, for some $i$. It follows that $Λ^2U_j = C$ for $j ≠ i$. This forces $l_i = A_1$ for $i = 1, 2$.

We use LiE notation [9] for irreducible modules, that is, we identify an irreducible representation with the coordinate vector of its highest weight, using the basis of fundamental weights.

**Lemma 13** Suppose $\hat{g} = A_1 ⊕ A_1 ⊕ l_3$ and $V = U_1 ⊗ U_2 ⊗ U_3$.

1. If $U_1$ and $U_2$ are copies of the standard representation of $A_1$, the algorithm tolerates for $U_3$ only one of the following types of representations:
   
   (a) $l_3 = A_n$, $U_3 = [1, 0, . . . , 0]$ or, $l_3 = A_3$, $U = [0, 1, 0]$, or $l_3 = A_1$, $U_3 = [2]$. In these cases, respectively, the algorithm terminates with $j = 2$, $g$ corresponding to $D_{n+3}$ with $α_{n+1}$-grading; $D_5^{(1)}$ with $α_2$-grading; $B_4^{(1)}$ with $α_2$-grading;

   (b) $l_3 = D_n$, $U_3 = [1, 0, . . . , 0]$; the algorithm terminates with $j = 2$, $g$ corresponding to $α_2$-graded $D_{n+2}$;

   (c) $l_3 = B_n$, $U_3 = [1, 0, . . . , 0]$; the algorithm terminates with $j = 2$, $g$ corresponding to $α_2$-graded $B_{n+2}$.

2. If $U_1 = [2]$, and $U_2 = [1]$, the algorithm aborts with $j = 2$ unless $U_3$ is the standard representation of $A_1$.

3. If either $U_1$ or $U_2$ is $[n]$, $n > 2$, the algorithm aborts with $j = 2$.

**Proof**

1. Suppose $U_1$ and $U_2$ are copies of the standard representation of $A_1$. We have

   $Λ^2V = C ⊗ S^2U_2 ⊗ S^2U_3 ⊗ S^2U_1 ⊗ C ⊗ S^2U_3 ⊗ S^2U_1 ⊗ S^2U_2 ⊗ Λ^2U_3 ⊗ C ⊗ C ⊗ Λ^2U_3$.

   Note that $S^2U_i = adj A_1$, for $i = 1, 2$.

   If $Λ^2U_3$ is not irreducible, the algorithm aborts with $j = 1$: it only tolerates a reducible $T_i^c$ if $T_i^c$ is the adjoint representation of $\hat{g}$, in particular, there can be no more than one irreducible component of $T_i^c$ per simple summand of $\hat{g}$. This means $l_3 ≠ C_n$.

   Suppose $Λ^2U_3$ is irreducible. If $S^2U_3$ has more than one nontrivial component, the algorithm aborts with $j = 1$: in this case, $C ⊗ adj A_1 ⊗ S^2U_3$ contributes to $T_i^c$ something other than the adjoint representation of $A_1$,
the net effect being $T_1^c$ is neither irreducible nor the adjoint representation of $\mathfrak{g}$. This guarantees that $t_3 \neq E_n$.

We can restrict attention to representations of type $A$, $B$, and $D$ algebras with $\Lambda^2U_3$ irreducible, and $S^2U_3$ with no more than one nontrivial irreducible component.

(a) Let $t_3 = A_n$. If $U_3 = [1, 0, \ldots, 0]$, it is easy to verify that the algorithm terminates successfully with $j = 2$. Suppose $U_3 = [0, 1, 0, \ldots, 0]$. If $n \geq 3$, $S^2U_3 = [0, 2, 0, \ldots, 0] + [0, 0, 0, 1, 0, \ldots, 0]$ so when $n > 3$, the algorithm aborts with $j = 1$. When $n = 3$, the second representation is $\mathbb{C}$. In this case, $\Lambda^2U_3 = \text{adj} A_3$. The algorithm tolerates this case, designates $V_2 = A_1 \oplus A_1 \oplus A_3$, and thus terminates successfully at $j = 2$.

If $U_3$ is a higher exterior power of the standard representation of $A_n$, $S^2U_3$ has more than one nontrivial irreducible submodule so the algorithm aborts in these cases with $j = 1$. The same observation applies to higher degree cominuscule representations of $A_n$, except when $U_3 = \text{adj} A_1$. Here, the algorithm yields $V_2 = A_1 \oplus A_1 \oplus A_1$, terminating successfully with $j = 2$.

(b) Let $t_3 = D_n$. If $U_3 = [1, 0, \ldots, 0]$, $S^2U_3 = [2, 0, \ldots, 0] + \mathbb{C}$ and $\Lambda^2U_3 = \text{adj} D_n$. The algorithm designates $V_2 = A_1 \oplus A_1 \oplus D_n$, terminating successfully with $j = 2$.

Suppose $U_3 = \text{adj} D_n$. In this case, $S^2U_3$ has more than one nontrivial component so the algorithm aborts with $j = 1$.

If $n = 4$, we can take $U_3 = [0, 0, 1, 0]$ because then $S^2U_3 = [0, 0, 2, 0] + \mathbb{C}$ and $\Lambda^2U_3 = \text{adj} D_4$. If $U_3$ is a half-spin representation when $n > 4$, there is a second nontrivial component of $S^2U_3$ which makes $T_1^c$ neither irreducible nor adjoint. In these cases, the algorithm aborts with $j = 1$.

If $U_3$ is a higher degree cominuscule representation of $D_n$, $S^2U_3$ has more than one nontrivial component and the algorithm aborts with $j = 1$.

(c) The cases where $U_3$ is a representation of $B_n$ are similar to those where $U_3$ is a fundamental representation of $D_n$. We leave details to the reader.

2. Now suppose $U_1 = [2]$ and $U_2 = [1]$. In this case $S^2U_1 = [4] + \mathbb{C}$ so

$$T_1^c = \mathbb{C} \otimes \mathbb{C} \otimes S^2U_3 \otimes \mathbb{C} \otimes \text{adj} A_1 \otimes \Lambda^2U_3 \otimes \text{adj} A_1 \otimes \mathbb{C} \otimes \Lambda^2U_3.$$

The algorithm aborts unless this is the adjoint representation of $\mathfrak{g}$. That happens precisely when $\Lambda^2U_3$ is trivial and $S^2U_3$ is the adjoint representation of the third simple component of $\mathfrak{g}$. That is, $U_3$ must be the standard representation of $A_1$. 

22
3. Suppose $U_1$ and $U_2$ are copies of $\text{adj } A_1$. In this case, $\mathcal{T}^c_1$ has at least three irreducible components, one of which is $\text{adj } A_1 \otimes \text{adj } A_1 \otimes \Lambda^2 U_3$. This forces the algorithm to abort. Higher symmetric powers of the standard representation of $A_1$ yield similar results. \hfill \Box

Next we consider what happens when $\mathcal{T}^c_1 = \Lambda^2 U_1 \otimes \Lambda^2 U_2 \otimes \Lambda^2 U_3$. As per remarks preceding the theorem, $t_i$ is type $A$ for all $i$ and $U_i = [1, 0, \ldots, 0]$. In all these cases, the algorithm advances to $j = 2$ with $V_2 = \Lambda^2 U_1 \otimes \Lambda^2 U_2 \otimes \Lambda^2 U_3$. Let $C_{1,i}(U_j)$ be the Cartan product of $U_j$ and $\Lambda^i U_j$. We have

$$V_1 \otimes V_2 = (C_{1,2}(U_1) \oplus \Lambda^3 U_1) \otimes (C_{1,2}(U_2) \oplus \Lambda^3 U_2) \otimes (C_{1,2}(U_3) \oplus \Lambda^3 U_3),$$

so that

$$\mathcal{T}^c_2 = C_{1,2}(U_1) \otimes \Lambda^3 U_2 \otimes \Lambda^3 U_3 \oplus \Lambda^3 U_1 \otimes \Lambda^3 U_2 \otimes C_{1,2}(U_3) \oplus \Lambda^3 U_1 \otimes \Lambda^3 U_2 \otimes \Lambda^3 U_3.$$ 

The algorithm directs that one of the following happens.

1. If $\mathcal{T}^c_2 = \{0\}$ the algorithm terminates successfully with $j = 2$.

2. If $\mathcal{T}^c_2$ is irreducible, the algorithm advances to $j = 3$ with $V_3 = \mathcal{T}^c_2$.

3. Suppose $\mathcal{T}^c_2 \cong \text{adj } \mathfrak{g} \oplus \mathbb{C}, V_3 = \text{adj } \mathfrak{g}$ and the algorithm terminates successfully with $j = 3$.

**Lemma 14** $\mathcal{T}^c_2 = \{0\}$ if and only if $U_1$ and $U_2$ are standard representations of $A_1$. In this case, the algorithm terminates successfully with $\mathfrak{g}$ corresponding to $\alpha_{n_3+1}$-graded $D_{n_3+3}$.

**Proof** We work under the assumption that $n_1 \leq n_2 \leq n_3$ so $\mathcal{T}^c_2 = \{0\}$ precisely when $\Lambda^3 U_i = \{0\}$ for $i = 1, 2$, which happens only if $n_i = 1$, for $i = 1, 2$. \hfill \Box

Now we work under the assumption that the algorithm has advanced to $j = 2$.

**Lemma 15** $\mathcal{T}^c_2$ is irreducible only if $U_1 = [1]$ and $n_2 > 1$.

**Proof** $\mathcal{T}^c_2$ is irreducible precisely when $\Lambda^3 U_i = \{0\}$ for exactly one $i$, thus when $i = 1$. This means $t_{n_1} = A_1$. \hfill \Box

**Lemma 16** $\mathcal{T}^c_2 = \text{adj } \mathfrak{g} \oplus \mathbb{C}$ if and only if $U_1 = U_2 = U_3$ is the standard representation of $A_2$. In this case the algorithm terminates successfully with $j = 3$, and $\mathfrak{g}$ corresponding to $\alpha_1$-graded $E_6^{(1)}$.

**Proof** $\mathcal{T}^c_2 = \text{adj } \mathfrak{g} \oplus \mathbb{C}$ only if $\Lambda^3 U_i = \mathbb{C}$ for all $i$, which implies the result. \hfill \Box

We have proved the following.
Proposition 4  When \( V = U_1 \otimes U_2 \otimes U_3 \), the algorithm advances beyond \( j = 3 \) only if each \( U_i \) is the standard representation of \( A_{n_i} \) with \( n_1 = 1 \), and \( n_3 \geq n_2 \geq 2 \). In this case we have \( V_3 = U_1 \otimes \Lambda^3 U_2 \otimes \Lambda^3 U_3 \). \( \square \)

Suppose \( V_3 \) is irreducible. We have

\[
V_1 \otimes V_3 = ( \text{adj} \ A_1 \otimes \mathbb{C}) \otimes (C_{1,3}(U_2) \oplus \Lambda^4 U_2) \otimes (C_{1,3}(U_3) \oplus \Lambda^4 U_3).
\]

This gives us

\[
\mathcal{T}_3^c = \text{adj} \ A_1 \otimes \Lambda^4 U_2 \otimes \Lambda^4 U_3 \otimes C \otimes C_{1,3}(U_2) \otimes \Lambda^4 U_3 \oplus \\
\mathbb{C} \otimes \Lambda^4 U_2 \otimes C_{1,3}(U_3) \otimes \mathbb{C} \otimes \Lambda^4 U_2 \otimes \Lambda^4 U_3.
\]

The algorithm directs that one of the following happens.

1. If \( \mathcal{T}_3^c = \{0\} \) the algorithm terminates successfully with \( j = 3 \).
2. If \( \mathcal{T}_3^c \) is irreducible, the algorithm advances to \( j = 4 \) with \( V_4 = \mathcal{T}_3^c \).
3. If \( \mathcal{T}_4^c \cong \text{adj} \ \mathfrak{g} \oplus \mathbb{C}, V_4 = \text{adj} \ \mathfrak{g} \) and the algorithm terminates successfully with \( j = 4 \).

Lemma 17  Suppose the algorithm has advanced to \( j = 3 \). \( V_3 \) is irreducible if and only if \( U_1 \) is the standard representation of \( A_1 \), and each of \( U_2 \) and \( U_3 \) is the standard representation of \( A_2 \). In this case, the algorithm terminates successfully with \( j = 3 \) and \( \mathfrak{g} \) corresponding to \( \alpha_4 \)-graded \( E_6 \).

Proof  \( V_4 = \{0\} \) precisely when \( \Lambda^4 U_2 = \Lambda^4 U_3 = \{0\} \) which happens precisely when \( A_{n_i} = A_2, i = 2, 3 \). \( \square \)

Lemma 18  Suppose the algorithm has advanced to \( j = 4 \). \( V_4 \) is irreducible precisely when \( U_2 \) is the standard representation of \( A_2 \) and \( n_3 > 2 \).

Proof  \( V_4 \) is irreducible precisely when \( \Lambda^4 U_2 = \{0\} \) and \( \Lambda^4 U_3 \neq \{0\} \). The result follows from there. \( \square \)

Lemma 19  If the algorithm advances to \( j = 4 \), \( V_4 = \text{adj} \ \mathfrak{g} \) precisely when each of \( U_2, U_3 \) is the standard representation of \( A_3 \). In this case, the algorithm terminates successfully to produce \( \mathfrak{g} \) corresponding to \( \alpha_4 \)-graded \( E_7^{(1)} \).

Proof  \( V_4 = \text{adj} \ \mathfrak{g} \) precisely when \( \mathbb{C} \otimes \Lambda^4 U_2 \otimes \Lambda^4 U_3 \cong \mathbb{C}, \) that is, when \( \Lambda^3 U_2 \cong \Lambda^3 U_3 \cong \mathbb{C}, \) that is, when \( U_2 \) and \( U_3 \) are copies of the standard representation of \( A_3 \). \( \square \)

Suppose the algorithm has advanced to \( j = 4 \) and that \( V_4 = \mathbb{C} \otimes U_2 \otimes \Lambda^4 U_3 \) is irreducible, with \( U_2 \) the standard representation of \( A_2 \). Since \( V_1 \otimes V_4 = U_1 \otimes (S^2 U_2 \oplus U_2^*) \otimes (C_{1,4} \oplus \Lambda^5 U_3) \),

\[
\mathcal{T}_4^c = U_1 \otimes U_2^* \otimes \Lambda^5 U_3. \tag{5}
\]

This implies the following.
Lemma 20 When \( V = U_1 \otimes U_2 \otimes U_3 \), the algorithm cannot produce \( V_5 = \text{adj}^k \mathfrak{g} \).
\[ \square \]

Lemma 21 Suppose the algorithm advances to \( j = 4 \) with \( V_4 \) irreducible, \( \mathcal{T}_4^\mathfrak{c} = \{0\} \) precisely when \( U_1 \) is the standard representation of \( A_1 \), \( U_2 \) the standard representation of \( A_2 \), and \( U_3 \) the standard representation of \( A_3 \). In this case, the algorithm terminates successfully with \( j = 3 \) and \( \mathfrak{g} \) corresponding to \( \alpha_4 \)-graded \( E_7 \).

\[ \text{Proof} \ \mathcal{T}_4^\mathfrak{c} = \{0\} \text{ precisely when } \Lambda^3 U_3 = \{0\}, \text{ which happens if and only if } U_3 \text{ is the standard representation of } A_3. \ \square \]

Lemma 22 If \( V = U_1 \otimes U_2 \otimes U_3 \) and for \( i = 2, 3, 4, 5 \) \( V_i \) are defined and irreducible, then \( U_1 \) is the standard representation of \( A_1 \), \( U_2 \) the standard representation of \( A_2 \), and \( U_3 \) the standard representation of \( A_n \) for some \( n > 3 \). In this case, \( V_5 = U_1 \otimes U_2 \otimes \Lambda^6 U_3 \).

\[ \text{Proof} \ \text{The proof follows Eq. (5).} \ \square \]

Suppose the algorithm has advanced to \( j = 5 \) with \( V_5 \) irreducible, in particular, with \( n_3 > 3 \). We have
\[ V_1 \otimes V_5 = (\text{adj } A_1 \oplus \mathbb{C}) \otimes (\text{adj } A_2 \oplus \mathbb{C}) \otimes (C_{1,5}(U_3) \oplus \Lambda^6 U_3). \]

If \( n_3 = 4 \), \( C_{1,5}(U_3) = U_3 \) and \( \Lambda^6 U_3 = \{0\} \) but in any case, the following is never zero
\[ \mathcal{T}_5^\mathfrak{c} = \text{adj } A_1 \otimes \mathbb{C} \otimes \Lambda^6 U_3 \oplus \mathbb{C} \otimes \mathbb{C} \otimes C_{1,5}(U_3) \oplus \mathbb{C} \otimes \mathbb{C} \otimes \Lambda^6 U_3. \]

Lemma 23 \( V_6 \) is irreducible if and only if \( U_3 \) is the standard representation of \( A_4 \). In this case, the algorithm terminates successfully with \( j = 6 \) and \( \mathfrak{g} \) corresponding to \( \alpha_4 \)-graded \( E_8 \).

\[ \text{Proof} \ \text{If } n_3 > 4, \mathcal{T}_5^\mathfrak{c} \text{ has at least two irreducible components so the algorithm aborts with } j = 5 \text{ unless } \mathcal{T}_5^\mathfrak{c} = \text{adj } \mathfrak{g} \oplus \mathbb{C}. \text{ This happens, in turn, only if } \Lambda^6 U_3 = \mathbb{C}, \text{ that is, if and only if } U_3 \text{ is the standard representation of } A_5. \ \square \]

The following summarizes what the algorithm produces when \( \mathfrak{g} \) has three simple components.

Proposition 5 When \( \mathfrak{g} \) has more than two simple components the algorithm either aborts with \( j \leq 5 \), or it terminates successfully with \( j \leq 6 \) and \( \mathfrak{g} \) corresponding to one of the following types: \( \alpha_2 \)-graded \( B_6^{(1)} \), \( \alpha_2 \)-graded \( D_6 \), \( \alpha_2 \)-graded \( D_6^{(1)} \), \( \alpha_4 \)-graded \( E_6 \), or \( \alpha_4 \)-graded \( E_8^{(1)} \). \[ \square \]
5.2 $V = U_1 \otimes U_2$

$U_i$ is a generalized cominuscule representation of finite type $I_i$, $V = U_1 \otimes U_2$, and $\mathfrak{g} = l_1 \oplus l_2$. We have $\Lambda^2 V = \Lambda^2 U_1 \otimes S^2 U_2 \oplus S^2 U_1 \otimes \Lambda^2 U_2$. The case where $\mathcal{T}_I^c = \{0\}$ is part of the Minuscule Algorithm [7]. To complete our accounting we note that $\mathcal{T}_I^c = \{0\}$ if and only if $S^2 U_i$ is irreducible for $i = 1, 2$; that is, if and only if $U_1$ and $U_2$ are standard representations of type $A$ algebras. In such a case, the algorithm terminates successfully with $j = 1$ and $\mathfrak{g}$ corresponding to $\alpha_k$-graded $A_n$, $1 < k < n$.

$\mathcal{T}_I^c$ is irreducible if and only if $\Lambda^2 U_i$ and $S^2 U_1$ are irreducible and $S^2 U_2$ has two irreducible components. $U_1$ must be a standard representation of a type $A$ algebra since all other generalized cominuscule representations of finite type algebras have a reducible symmetric square. For the duration of this discussion, let $l_1 = A_n$ and $U_1 = [1, 0, \ldots, 0]$. Choosing $U_2$ from the following list guarantees that $\mathcal{T}_I^c$ is irreducible. The list is exhaustive.

1. $l_2 = A_m$, $U_2 = [2, 0, \ldots, 0]$;
2. $l_2 = A_m$, $U_2 = [0, 1, 0, \ldots, 0]$;
3. $l_2 = B_m$ or $D_m$, $U_2 = [1, 0, \ldots, 0]$;
4. $l_2 = D_5$, $U_2 = [0, 0, 0, 0, 1]$;
5. $l_2 = E_6$, $U_2 = [1, 0, 0, 0, 0, 0]$.

The next sequence of lemmas analyzes these cases in order. When we encounter twisted affine Lie algebras, we number the simple roots as in [5].

**Lemma 25** Let $l_2 = A_m$ and $U_2 = [2, 0, \ldots, 0]$. The algorithm terminates successfully if and only if one of the following holds.

1. $m = 1$, in which case $j = 2$ and $\mathfrak{g}$ corresponds to $\alpha_{n+1}$-graded $B_{n+2}$;
2. $n = 1$ or $2$ and $m = 2$, in which case $j = 2$ and $\mathfrak{g}$ corresponds to $\alpha_2$-graded $F_4$ or $F_4^{(1)}$, respectively; or
3. $n = 1$ and $m = 3$, in which case $j = 4$ and $\mathfrak{g}$ corresponds to $\alpha_3$-graded $E_6^{(2)}$.

Otherwise, the algorithm aborts with $j \leq 3$.

**Proof** Under the hypotheses we have $S^2 U_2 = [4, 0, \ldots, 0] \oplus [0, 2, 0, \ldots, 0]$. If $m = 1$, then $[0, 2, 0, \ldots, 0] = [0] = \mathbb{C}$. Note that $\mathcal{T}_I^c = \Lambda^2 U_1 \otimes [0, 2, 0, \ldots, 0]$. This is always nonzero and irreducible so the algorithm always advances to $j = 2$. Next consider $V_1 \otimes V_2$.

If $m = 1$, $V_2 = \Lambda^2 U_1 \otimes \mathbb{C}$ so that $V_1 \otimes V_2 = (C_{1,2}(U_1) \oplus \Lambda^3 U_1) \otimes (U_2 \otimes \mathbb{C})$. The algorithm terminates successfully with $j = 2$ as $\mathcal{T}_I^c = \{0\}$. 

26
For \( m \geq 2 \), \( V_2 \otimes [0, 2, 0, \ldots, 0] = [2, 2, 0, \ldots, 0] \oplus [1, 1, 1, 0, \ldots, 0] \oplus [0, 0, 2, 0, \ldots, 0] \).
This gives us
\[
T_2^c = C_{1, 2}(U_1) \otimes [0, 0, 2, 0, \ldots, 0] \oplus \Lambda^3 U_1 \otimes [1, 1, 1, 0, \ldots, 0]
\oplus \Lambda^3 U_1 \otimes [0, 0, 2, 0, \ldots, 0].
\] (6)

Expression (6), always nonzero, is irreducible precisely when \( \Lambda^3 U_1 = \{0\} \),
that is, when \( A_3 = A_1 \), so that \( C_{1, 2}(U_1) = U_1 \). In this case, the algorithm
designates \( V_3 = U_1 \otimes [0, 0, 2, 0, \ldots, 0] \), which becomes \( U_1 \otimes C \) when \( m = 2 \). For
\( m > 2 \),
\[ V_1 \otimes V_3 = (\text{adj } A_1 \oplus C) \otimes [2, 0, 2, 0, \ldots, 0] \oplus [1, 0, 1, 0, \ldots, 0] \oplus [0, 0, 2, 0, \ldots, 0], \]
and for \( m = 2 \),
\[ V_1 \otimes V_3 = (\text{adj } A_1 \oplus C) \otimes U_2. \]
This leaves us with
\[
T_3^c = \text{adj } A_1 \otimes [0, 0, 0, 2, 0, \ldots, 0] \oplus C \otimes [1, 0, 1, 0, \ldots, 0]
\oplus C \otimes [0, 0, 2, 0, \ldots, 0],
\] (7)
when \( m > 2 \), and \( T_3^c = \{0\} \) when \( m = 2 \). In the latter case, the algorithm
terminates successfully with \( j = 2 \).

As long as \( m > 2 \), the expression in (7) is not irreducible. It is equivalent to
\( \text{adj } A_1 \oplus \text{adj } A_3 \oplus C \) if and only if \( m = 3 \). In this case, the algorithm terminates
successfully with \( j = 4 \), \( V_4 = \text{adj } A_1 \oplus \text{adj } A_3 \). If \( m > 3 \), the algorithm aborts
with \( j = 3 \).

Next we consider when (6) is \( \text{adj } A_1 \oplus C \). We need \( C_{1, 2}(U_1) = \text{adj } A_2, \]
\( [0, 0, 2, 0, \ldots, 0] = C \), and \( \Lambda^3 U_1 = C \). This happens precisely when \( n = m = 2 \),
giving us a successful termination of the algorithm with \( j = 3 \). \( \square \)

**Lemma 26** Let \( l_2 = A_m \), and \( U_2 = [0, 1, 0, \ldots, 0] \). The algorithm terminates
successfully if and only if one of the following holds.

1. \( m = 3 \), in which case case \( j = 2 \) and \( g \) corresponds to \( \alpha_{n+1} \)-graded \( D_{n+4} \);
2. \( 1 \leq n \leq 4 \) and \( m = 4 \); in these cases, \( j = 2, 3, 5, 5 \) respectively and \( g \)
corresponds to \( \alpha_5 \)-graded \( E_\ell \), \( \ell = 6, 7, 8 \), or \( \alpha_5 \)-graded \( E_8^{(1)} \), respectively;
3. \( n = 1 \) or \( 2 \), and \( m = 5 \); in these cases, \( j = 3 \) and \( g \) corresponds to
\( \alpha_3 \)-graded \( E_7 \) or \( \alpha_3 \)-graded \( E_7^{(1)} \), respectively;
4. \( n = 1 \), and \( m = 6 \) or \( 7 \); in these cases, \( j = 4 \), and \( g \) corresponds to
\( \alpha_3 \)-graded \( E_8 \) or \( \alpha_3 \)-graded \( E_8^{(1)} \) respectively.

In all other cases, the algorithm aborts with \( j \leq 4 \).
Proof} We have $A^2 U_2 = [1, 0, 1, 0, \ldots, 0]$ and $S^2 U_2 = [0, 2, 0, \ldots, 0] + [0, 0, 0, 1, 0, \ldots, 0]$ so that $V_2 = [0, 1, 0, \ldots, 0] \otimes [0, 0, 0, 1, 0, \ldots, 0]$ and

$$V_1 \otimes V_2 = ([1, 1, 0, \ldots, 0] + [0, 0, 1, 0, \ldots, 0]) \otimes ([0, 1, 0, 1, 0, \ldots, 0] + [0, 0, 0, 0, 1, 0, \ldots, 0]).$$

This gives us $T_2^c = [1, 1, 0, \ldots, 0] \otimes [0, 0, 0, 0, 1, 0, \ldots, 0] + [0, 0, 1, 0, \ldots, 0] \otimes [0, 0, 0, 0, 1, 0, \ldots, 0].$ (8)

When $n = 2$ and $m = 5$, $T_2^c \equiv \text{adj} \ g + \mathbb{C}$ so the algorithm terminates successfully with $j = 2$. All other combinations of $n$ and $m$ with $n > 1$ and $m > 4$ yield $V$ inadmissible, the algorithm aborting here with $j = 2$. When $m = 3$, $V_2 = [0, 1, 0, \ldots, 0] \otimes \mathbb{C}$ for arbitrary $n$. In these cases, $V_1 \otimes V_2 = T_2$ and the algorithm terminates successfully with $j = 2$.

Restrict now to $n = 1$, $m > 3$. Eq. (8) becomes $[1] \otimes [0, 0, 0, 0, 1, 0, \ldots, 0]$, which is zero when $m = 4$. Thus, $n = 1$ and $m = 4$ yield a successful run of the algorithm terminating with $j = 2$. Taking $m = 5$, we get $V_3 = [1] \otimes \mathbb{C}$. Continuing, $V_1 \otimes V_3 = ([2] + \mathbb{C}) \otimes U_3$ so that $T_3^c$ is zero and the algorithm terminates successfully with $j = 3$. Letting $m = 6$, we get $V_2 = [1] \otimes [0, 0, 0, 0, 0, 1, 0]$ so that $V_1 \otimes V_3 = ([2] + \mathbb{C}) \otimes ([0, 1, 0, 0, 0, 1] + [1, 0, 0, 0, 0, 0])$ and $V_4 = \mathbb{C} \otimes [1, 0, 0, 0, 0, 0]$. Then $V_1 \otimes V_4 = [1] \otimes ([1, 1, 0, 0, 0, 0] + [0, 0, 1, 0, 0, 0])$ which implies that $T_4^c = \{0\}$, thus, that the algorithm terminates successfully with $j = 4$. If $m = 7$, $V_3 = [1] \otimes [0, 0, 0, 0, 0, 1, 0]$ and $V_1 \otimes V_3 = ([2] + \mathbb{C}) \otimes ([0, 1, 0, 0, 0, 0, 1] + [1, 0, 0, 0, 0, 1] + \mathbb{C})$, giving us $T_3^c = [2] \otimes \mathbb{C} + \mathbb{C} \otimes [1, 0, 0, 0, 0, 0, 1] + \mathbb{C} \otimes \mathbb{C}$. Then $V_4 \equiv \text{adj} \ g$ and the algorithm terminates successfully with $j = 4$. If $n = 1$ and $m > 7$, the adjoint representation of $t_2$ does not appear in $T_3^c$ so the algorithm aborts at $j = 3$.

Next take $n = 3$. We need only consider $m = 4$ and $m = 5$, each $T_2^c = \mathbb{C} \otimes [1, 0, 0, 0]$ giving us $V_1 \otimes V_3 = [1] \otimes ([1, 1, 0, 0, 0] + [0, 0, 1, 0, 0])$. Since $T_3^c = \{0\}$, the algorithm terminates successfully with $j = 3$.

Now reconsider Eq. (8) when $n > 2$, $m > 3$. Since $V$ is inadmissible when $m \geq 5$, we only need consider $m = 4$. Then $T_2^c = [0, 0, 1, 0, \ldots, 0] \otimes [0, 0, 0, 1, 0, 0] \otimes [0, 0, 1, 0]$ so that $T_3^c = [0, 0, 0, 1, 0, \ldots, 0] \otimes [0, 0, 1, 0]$. The algorithm advances to $j = 4$ and we have

$$V_1 \otimes V_4 = ([1, 0, 0, 1, 0, \ldots, 0] + [0, 0, 0, 0, 1, 0, \ldots, 0]) \otimes ([0, 1, 1, 0] + [1, 0, 0, 1] + \mathbb{C}).$$

If $n > 4$, the algorithm aborts with $j = 4$. When $n = 4$, $T_4^c = \text{adj} \ g + \mathbb{C}$ so the algorithm advances successfully with $j = 5$. When $n = 3$, the algorithm advances with $V_4 = [1, 0, 0] \otimes \mathbb{C}$. Then $V_1 \otimes V_4 = T_4$ so the algorithm terminates successfully with $j = 4$. □

**Lemma 27** Let $t_2 = B_m$ or $D_m$ and $U_2 = [1, 0, \ldots, 0]$. The algorithm terminates successfully with $j = 2$, $g$ corresponding respectively to $\alpha_{n+1}$-graded $B_{n+m+1}$ or $D_{n+m+1}$.
Proof If $U_2$ is the standard representation of $B_m$, then $\Lambda^2 U_2 = \text{adj} B_m$ and $S^2 U_2 = [2,0,\ldots,0] \oplus \mathbb{C}$. Thus,

$$\mathcal{T}_1^c = \Lambda^2 U_1 \otimes \mathbb{C},$$

which is always irreducible. The algorithm designates $V_2 = \Lambda^2 U_1 \otimes \mathbb{C}$ and we have

$$V_1 \otimes V_2 = (C_{1,2}(U_1) \oplus \Lambda^3 U_1) \otimes U_2.$$

Since $\mathcal{T}_2^c = \{0\}$, the algorithm terminates successfully with $j = 2$.

The argument applies exactly as stated if $U_2$ is the standard representation of $D_m$. □

Lemma 28 Let $l_2 = D_5$ and $U_2 = [0,0,0,0,1]$. When $n = 1,2$ or 3, the algorithm terminates successfully, with $j = 2$ when $n = 1$, and $j = 4$ when $n = 2$ or 3. $g$ corresponds respectively to $\alpha_6$-graded $E_7$, $\alpha_6$-graded $E_8$, or $\alpha_6$-graded $E_8^{(3)}$. Otherwise, the algorithm aborts with $j = 3$.

Proof Under our hypotheses, $S^2 U_2 = [0,0,0,0,2] \oplus [1,0,0,0,0] \oplus [1,0,0,0,1] \oplus [0,0,0,0,0]$ so that $\mathcal{T}_1^c$ is always irreducible and the algorithm always advances to $j = 2$. This gets us

$$V_2 = \mathcal{T}_1^c = \Lambda^2 U_1 \otimes [1,0,0,0,0]$$

and

$$V_1 \otimes V_2 = (C_{1,2}(U_1) \oplus \Lambda^3 U_1) \otimes ([1,0,0,0,1] \oplus [0,0,0,1,0]).$$

Note that

$$\mathcal{T}_2^c = \Lambda^3 U_1 \otimes [0,0,0,1,0]$$

is zero if $n = 1$, irreducible otherwise. We see, then, that when $n = 1$, the algorithm terminates successfully with $j = 2$. If $n > 1$, the algorithm advances to $j = 3$ with $V_3 = \Lambda^3 U_1 \otimes [0,0,0,1,0]$. This gives us

$$V_1 \otimes V_3 = (C_{1,3}(U_1) \oplus \Lambda^4 U_1) \otimes ([0,0,0,1,1] \oplus \text{adj} D_5 \oplus \mathbb{C}),$$

so that

$$\mathcal{T}_3^c = C_{1,3}(U_1) \otimes \mathbb{C} \oplus \Lambda^4 U_1 \otimes \text{adj} D_5 \oplus \Lambda^4 U_1 \otimes \mathbb{C}.$$

The expression in (9) is irreducible if and only if $\Lambda^4 U_1 = \{0\}$, that is, if and only if $A_n = A_2$, in which case $V_4 = U_1$. Continuing, we have

$$V_1 \otimes V_4 = (S^2 V_1 \oplus \Lambda^2 U_1) \otimes V_2$$

so that $\mathcal{T}_4^c = \{0\}$, and the algorithm terminates successfully with $j = 4$.

Next consider that the expression in (9) is equivalent to $\text{adj}^k g \oplus \mathbb{C}$ if and only if $\Lambda^4 V_1 = \mathbb{C}$, which is true if and only if $A_n = A_3$. In this case, the algorithm terminates successfully with $j = 4$.

For all values of $n > 3$, the expression in (9) forces the algorithm to abort with $j = 3$. □
Lemma 29  Let $l_2 = E_6$, $U_2 = [1, 0, 0, 0, 0, 0]$. The algorithm terminates successfully if and only if $n = 1$ or 2, in which case $j = 3$ and $\mathfrak{g}$ corresponds respectively to $\alpha_7$-graded $E_8$ or $\alpha_7$-graded $E_8^{(1)}$. Otherwise, the algorithm aborts with $j = 2$.

Proof  Under the hypotheses, say $U_2 = [1, 0, 0, 0, 0, 0]$. We have
\[
\mathcal{T}_1^e = \Lambda^2 U_1 \otimes [0, 0, 0, 0, 0, 1]
\]
so that the algorithm always advances to $j = 2$. Then
\[
U_1 \otimes U_2 = (C_{1,2}(U_1) \oplus \Lambda^3 U_1) \otimes ([1, 0, 0, 0, 0, 1] \oplus \text{adj} E_6 \oplus \mathbb{C}),
\]
leaving us with
\[
\mathcal{T}_2^e = C_{1,2}(U_1) \otimes \mathbb{C} \oplus \Lambda^3 U_1 \otimes \text{adj} E_6 \oplus \Lambda^3 U_1 \otimes \mathbb{C}. \tag{10}
\]
This is irreducible if and only if $\Lambda^3 U_1 = \{0\}$, that is, if and only if $A_n = A_1$. In this case, $V_4 = U_1 \otimes \mathbb{C}$ and we continue, to find $V_4 \otimes V_4 = \mathcal{T}_4$. This is a successful termination of the algorithm with $j = 3$.

The expression in (10) is equivalent to $\text{adj}^k \mathfrak{g} \oplus \mathbb{C}$ if and only if $\Lambda^3 U_1 = \mathbb{C}$, that is, if and only if $A_n = A_2$. In this case, the algorithm terminates successfully with $j = 3$.

In all other cases, the expression in (10) forces the algorithm to abort with $j = 2$. □

This exhausts the cases where $\mathcal{T}_1^e$ irreducible. Next, we consider conditions on $V = U_1 \otimes U_2$ that force $\mathcal{T}_1^e \cong \text{adj}^k \mathfrak{g} + \mathbb{C}$. Since
\[
\Lambda^2 V = \Lambda^2 U_1 \otimes S^2 U_2 \oplus S^2 U_1 \otimes \Lambda^2 U_2,
\]
each summand of $\Lambda^2 V$ would have the form $W + \text{ad} \mathfrak{t}_i \otimes \mathbb{C}$, where $W$ is itself a tensor product, either of something with $\text{ad} \mathfrak{t}_i$ or something with $\mathbb{C}$. Note in particular that neither $\Lambda^2 U_i$ nor $S^2 U_i$ can have the form $\text{ad} \mathfrak{t}_i \otimes \mathbb{C}$. Note further that $S^2 U_i \neq \mathbb{C}$ and that $\Lambda^2 U_i = \mathbb{C}$ if and only if $U_i$ is the standard representation of $A_1$, in which case $S^2 U_i = \text{ad} A_1$. These allow us to limit the criteria determining $U_1$ and $U_2$ to the following.

1. $\Lambda^2 U_i = \text{ad} \mathfrak{t}_i$, $S^2 U_i = Z \oplus \mathbb{C}$, where $Z$ is irreducible;

2. $U_i$ is the standard representation of $A_1$, $S^2 U_2 = Z + \text{ad} \mathfrak{t}_2$, and $\Lambda^2 U_2 = W + \mathbb{C}$, where $Z$ and $W$ are irreducible.

What follows is an complete list of generalized cominuscule representations $U$ that satisfy (1). (LiE [9] helps verify that the list is exhaustive.) Taking $U_1$, $U_2$ to be any pair from the list, we get $\mathcal{T}_1^e \cong \text{adj}^k \mathfrak{g} + \mathbb{C}$.

1. $\mathfrak{t} = A_1$, $U = [2]$;
2. \( \mathfrak{l} = A_3, \ U = [0,1,0] \);
3. \( \mathfrak{l} = B_n, \ U = [1,0,\ldots,0] \);
4. \( \mathfrak{l} = D_n, \ U = [1,0,\ldots,0] \).

There are sixteen pairs we can choose. With \( V = U_1 \otimes U_2 \), the algorithm terminates successfully with \( j = 2 \) and \( \mathfrak{g} \) corresponding to one of the following types of lsn-graded affine algebras: \( \alpha_k \)-graded \( D^{(2)}_{n+1} \), \( k \in \{1,\ldots,n-1\} \); \( \alpha_k \)-graded \( D^{(1)}_{n} \), \( k \in \{3,\ldots,n-3\} \); \( \alpha_k \)-graded \( B^{(1)}_{n} \), \( k \in \{3,\ldots,n-1\} \). Further, any such lsn-graded algebra can be produced by the algorithm for some choice of \( U_1 \) and \( U_2 \) on this list.

Next, consider the following exhaustive list of generalized cominuscule rep-

\[ \text{resentations } U \text{ with } S^2 U = Z + \text{ ad } \mathfrak{l}, \text{ and } \Lambda^2 U = W + \mathbb{C}, \text{ where } Z \text{ and } W \text{ are irreducible.} \]

1. \( \mathfrak{l} = A_1, \ U = [3] \);
2. \( \mathfrak{l} = A_5, \ U = [0,0,1,0,0] \);
3. \( \mathfrak{l} = C_3, \ U = [0,0,1] \);
4. \( \mathfrak{l} = D_6, \ U = [0,0,0,0,1,0] \);
5. \( \mathfrak{l} = E_7, \ U = [0,0,0,0,0,0,1] \).

Taking \( V = U_1 \otimes U_2 \), with \( U_1 \) the standard representation of \( A_1 \), and \( U_2 \) from this list, we get \( \mathfrak{g} \) corresponding respectively to: \( \alpha_2 \)-graded \( G^{(1)}_2 \); \( \alpha_3 \)-graded \( E^{(1)}_6 \); \( \alpha_1 \)-graded \( E^{(1)}_7 \); \( \alpha_8 \)-graded \( E^{(1)}_8 \).

The following summarizes our accounting in case \( V = U_1 \otimes U_2 \). We omit cases that terminate successfully with \( j = 1 \).

**Proposition 6** Let \( U_1 \) be a generalized cominuscule representation for finite type \( \mathfrak{l}_1 \), \( U_2 \) a generalized cominuscule representation for finite type \( \mathfrak{l}_2 \). The algorithm either aborts with \( j \leq 4 \) or it terminates successfully with \( 1 < j \leq 4 \) and \( \mathfrak{g} \) corresponding to one of the following types of lsn-graded algebras: \( \alpha_k \)-graded \( B_{n_k}, \ 1 < k < n-1 \); \( \alpha_k \)-graded \( B^{(1)}_{n_k}, \ 2 < k < n-1 \); \( \alpha_k \)-graded \( D_{n_k}, \ 1 < k < n-2 \); \( \alpha_k \)-graded \( D^{(1)}_{n_k}, \ 2 < k < n-2 \); \( \alpha_k \)-graded \( D^{(2)}_{n_k+1}, \ 0 < k < n \); \( \alpha_k \)-graded \( E_{n_k}, \ \alpha_k \)-graded \( E^{(1)}_{n_k}, \ k = 3 \) or \( 5 \leq k \leq n-1 \); \( \alpha_2 \)-graded \( E^{(1)}_6 \); \( \alpha_1 \)-graded \( E^{(1)}_7 \); \( \alpha_8 \)-graded \( E^{(1)}_8 \); \( \alpha_3 \)-graded \( E^{(2)}_6 \); \( \alpha_2 \)-graded \( F^{(1)}_4 \); \( \alpha_k \)-graded \( G^{(1)}_2 \). \( \square \)

### 5.3 \( \mathfrak{k} \) Simple

Here we detail the cases where \( \mathcal{T}^c_k \) is irreducible. Since \( \mathfrak{k} \) is simple, the irreducible cases subsume the cases where \( \mathcal{T}^c_k \cong \text{ adj } \mathfrak{k} \).

\( \mathcal{T}^c_k \) is zero precisely when \( \Lambda^2 V \) is irreducible. We list of these cases to have a complete account.

---

31
We list the representations $V_j$ the algorithm terminates successfully with $V_j$ at lemmas that follow.

1. $k \otimes V_1 = 2$. We note these in the list and analyze the rest of the cases through the lemmas.

2. $k \otimes V_2 = 2$.

3. $k \otimes V_3 = 2$. We note these in the list and analyze the rest of the cases through the lemmas.

4. $k \otimes V_4 = 2$.

5. $k \otimes V_5 = 2$.

6. $k \otimes V_6 = 2$.

7. $k \otimes V_7 = 2$.

8. $k \otimes V_8 = 2$.

$\mathcal{T}_c^1$ is irreducible precisely when $\Lambda^2 V$ has exactly two irreducible components.

We list the representations $V$ with this property. In cases where $\mathcal{T}_c^1 = \text{adj} \ k$, the algorithm terminates successfully with $j = 2$. In cases where $\mathcal{T}_c^1 = \mathbb{C}$, $V_1 \otimes V_2$ is irreducible, also resulting in a successful termination of the algorithm at $j = 2$. We note these in the list and analyze the rest of the cases through the lemmas that follow.

1. $k \otimes V_1 = 2$.

2. $k \otimes V_2 = 2$.

3. $k \otimes V_3 = 2$.

4. $k \otimes V_4 = 2$.

5. $k \otimes V_5 = 2$.

6. $k \otimes V_6 = 2$.

7. $k \otimes V_7 = 2$.

8. $k \otimes V_8 = 2$.

9. $k \otimes V_9 = 2$.

10. $k \otimes V_{10} = 2$.

11. $k \otimes V_{11} = 2$.

12. $k \otimes V_{12} = 2$.

13. $k \otimes V_{13} = 2$.

14. $k \otimes V_{14} = 2$.

15. $k \otimes V_{15} = 2$.

16. $k \otimes V_{16} = 2$.

17. $k \otimes V_{17} = 2$.

18. $k \otimes V_{18} = 2$.

19. $k \otimes V_{19} = 2$.

20. $k \otimes V_{20} = 2$.

21. $k \otimes V_{21} = 2$.

22. $k \otimes V_{22} = 2$.

23. $k \otimes V_{23} = 2$.

24. $k \otimes V_{24} = 2$.

25. $k \otimes V_{25} = 2$.

26. $k \otimes V_{26} = 2$.

27. $k \otimes V_{27} = 2$.

28. $k \otimes V_{28} = 2$.

29. $k \otimes V_{29} = 2$.

30. $k \otimes V_{30} = 2$.

31. $k \otimes V_{31} = 2$.

32. $k \otimes V_{32} = 2$.
Lemma 30 Let \( \mathfrak{g} = A_n \), \( n > 5 \), \( V_1 = [0, 0, 1, 0, \ldots, 0] \). Unless \( n = 6, 7 \) or \( 8 \), \( V \) is inadmissible and the algorithm aborts with \( j = 2 \). When \( n = 6 \) or \( 7 \), the algorithm terminates successfully with \( j = 2 \), and \( \mathfrak{g} \) corresponding respectively to \( \alpha_2 \)-graded \( E_7 \) or \( \alpha_2 \)-graded \( E_8 \). When \( n = 8 \), the algorithm terminates successfully with \( j = 2 \) and \( \mathfrak{g} \) corresponding to \( E_8^{(3)} \).

**Proof** When \( n = 6 \), \( T_{1}^c = [0, 0, 0, 0, 0, 1] \), \( V_1 \otimes V_2 = T_2 \), so the algorithm terminates successfully with \( j = 2 \). For \( n \geq 7 \), we have

\[
V_1 \otimes V_2 = [0, 0, 1, 0, 0, 1, 0, \ldots, 0] \oplus [0, 1, 0, 0, 0, 1, 0, \ldots, 0] \oplus [1, 0, 0, 0, 0, 0, 1, 0, \ldots, 0] \oplus [0, 0, 0, 0, 0, 0, 0, 1, 0, \ldots, 0].
\]

The algorithm thus aborts when \( n > 8 \). If \( n = 7 \), \( V_3 = [1, 0, 0, 0, 0, 0, 0, 0] \). Then

\[
V_1 \otimes V_3 = [1, 0, 1, 0, 0, 0, 0] \oplus [0, 0, 0, 1, 0, 0, 0],
\]

so that \( T_3^c = \{0\} \). The algorithm thus terminates successfully with \( j = 2 \). When \( n = 8 \), \( V_3 = \text{adj}_k \mathfrak{g} \) so the algorithm terminates successfully with \( j = 3 \). \( \square \)

Lemma 31 If \( \mathfrak{g} = A_n \), \( n > 7 \), and \( V_1 = [0, 0, 0, 1, 0, \ldots, 0] \), the algorithm aborts with \( j = 2 \).

**Proof** We have \( V_2 = [1, 0, 0, 0, 0, 0, 1, 0, \ldots, 0] \) so that when \( n > 7 \), \( V_1 \otimes V_2 \) has at least four nontrivial components. \( \square \)

Lemma 32 Let \( \mathfrak{g} = A_n \) and \( V_1 = [3, 0, \ldots, 0] \). The algorithm terminates successfully if and only if \( n = 1 \) or \( 2 \). When \( n = 1 \), \( j = 2 \) and \( \mathfrak{g} \) corresponds to \( \alpha_2 \)-graded \( G_2 \). When \( n = 2 \), \( j = 3 \) and \( \mathfrak{g} \) corresponds to \( \alpha_2 \)-graded \( D_4^{(3)} \).

If \( \mathfrak{g} = A_n \) and \( V_1 = [4, 0, \ldots, 0] \), the algorithm aborts unless \( n = 1 \). In this case, the algorithm terminates successfully with \( j = 2 \) and \( \mathfrak{g} \) corresponding to \( \alpha_1 \)-graded \( A_2^{(2)} \).

**Proof** If \( V_1 = [3, 0, \ldots, 0] \), then \( V_2 = [0, 3, 0, \ldots, 0] \) so that

\[
V_1 \otimes V_2 = [3, 3, 0, \ldots, 0] \oplus [2, 2, 1, 0, \ldots, 0] \oplus [1, 1, 2, 0, \ldots, 0] \oplus [0, 0, 3, 0, \ldots, 0].
\]

It is clear that the algorithm aborts with \( j = 2 \) when \( n > 2 \). When \( n = 2 \), \( V_3 = \text{adj}_k \mathfrak{g} \), and the algorithm terminates successfully with \( j = 3 \). When \( n = 1 \), \( V_2 = C \) so \( V_1 \otimes V_2 = T_2 \), giving us a successful termination of the algorithm with \( j = 2 \).

When \( k = 4 \), \( V_2 = [2, 3, 0, \ldots, 0] \) so that when \( n = 1 \), \( V_2 = \text{adj}_k \mathfrak{g} \) and the algorithm terminates successfully with \( j = 2 \). Otherwise, \( V_1 \otimes V_2 \) has eleven nontrivial components and the algorithm aborts. \( \square \)

Lemma 33 If \( \mathfrak{g} = A_n \), \( n > 3 \), and \( V = [0, 2, 0, \ldots, 0] \), then \( V \) is inadmissible, and the algorithm aborts with \( j = 2 \).
Proof We have $V_2 = T^c_1 = [1, 0, 1, 0, \ldots, 0]$ so that $V_1 \otimes V_2$ has more than five nontrivial irreducible components. □

Lemma 34 If $\mathfrak{g} = E_6$ with $V_1 = [2, 0, 0, 0, 0, 0]$, the algorithm aborts with $j = 2$.

Proof $V_2 = [0, 0, 1, 0, 1]$ so that $V_1 \otimes V_2$ has more than four nontrivial components, forcing the algorithm to abort. □

This completes the proof that there is an admissible representation associated to any affine or finite Lie algebra with lsn-grading and further, that these are the only admissible representations. In the next section, we verify that the algorithms actually produce Lie algebras with finite or affine type root systems.

6 Proof of the Algorithm, Part II: The Structure of $\mathfrak{g}$

Here we establish how the structure of an affine or finite type Lie algebra can be extended from $k^k$ to a $\mathbb{Z}$-graded vector space $\mathfrak{g}$ produced from the Affine or Finite Algorithm.

Suppose $\mathfrak{g}$ is constructed via the Affine Algorithm. Let $\gamma_1 = \sum_{i \neq k} c_i \alpha_i$ be the lowest weight of $\mathfrak{g}_1$. Define $a_k$ to be the least common denominator of the $c_i$s. Since weights of $\mathfrak{g}_1$ differ by elements of $Q^+$, $a_k$ is the least common denominator when we write any weight of $\mathfrak{g}_1$ as a linear combination of elements of $k^k \Pi$. Take $r = j/a_k$, where $j$ is minimal with $g_j \sim k^k g$. We claim that $j$ is a multiple of $a_k$: since $g_j \sim k^k g$, each $\alpha_i \in k^k \Pi$ must be a weight of $g_j$. The highest weight of $g_j$ has the form $j \lambda_1$ less $2j - 1$ positive roots of $\mathfrak{g}$. In particular, $j \lambda_1$ is in the root lattice of $\mathfrak{g}$ so $j/a_k$ must be a positive integer as claimed.

If the Finite Algorithm produces $\mathfrak{g}$, let $t_k = 1/(2 - (\lambda_1, \lambda_1))$ where $\lambda_1$ is the highest weight of $\mathfrak{g}_1$.

Let $B$ be the Killing form on $k^k$. Use $(.,.)$ to denote its rescaling as per §4.4, that is, the long roots associated to a simple component of $\mathfrak{g}$ have length $2/n$ where $n$ is the degree of the representation $\mathfrak{g}_1$ on that component of $k^k$.

The Affine Algorithm defines $\mathfrak{g}_0 = k^k \oplus \mathbb{C} + \mathbb{C}$. Identify one copy of $\mathbb{C}$ with $\mathbb{C}K$ and the other copy with $\mathbb{C}d_k$, where $Kx = 0$ for all $x \in \mathfrak{g}$ and $d_kx_t = tx$ if $x \in \mathfrak{g}_0$. In particular, define $[Kx] = [d_kx] = 0$ for all $x \in \mathfrak{g}_0$ so $\mathfrak{g}_0$ is a Lie algebra and $\mathfrak{g}$ is a $\mathfrak{g}_0$ module. Use Lemma 2 to extend the definition of $(.,.)|_{\mathfrak{h}}$ to all of $\mathfrak{h} := \mathfrak{h} + CK + \mathbb{C}d_k$. Decree that $K$ and $d_k$ are orthogonal to $x \in \mathfrak{g}$ if $x \neq \mathfrak{g}_0$. Let $\nu: \mathfrak{h} \rightarrow \mathfrak{h}^*$ be the isomorphism determined by the nondegenerate form on $\mathfrak{h}$. Define $\delta := \nu(K)$, $\Lambda_k := \nu(d_k)$, and $\alpha_k := \gamma_1 + (1/a_k) \delta = -\lambda_1 + (1/a_k) \delta$. 34
This puts $\alpha_k$ in $\mathfrak{h}^*$ with $(\alpha_k, \alpha_i) = (\gamma_1, \alpha_i) = -(\lambda_{-1}, \alpha_i)$ for all $\alpha_i \in \mathfrak{k}$. Note that $\|\alpha_k\|^2 = 2$. It is easy to verify that the net effect of all this on the roots of $\mathfrak{g}$ is to append a node representing $\alpha_k$ to $\mathcal{D}$ as it represents the module $\mathfrak{g}_{-1}$: $\alpha_k$ is long and shares $n$ edges with any node of $\mathcal{D}$ labeled $n$. Identify $\mathfrak{g}_{\alpha_k}$ with the $\gamma_1$ weight space of $\mathfrak{g}_1$ and $\mathfrak{g}_{-\alpha_k}$ with the $\lambda_{-1}$ weight space of $\mathfrak{g}_{-1}$.

The Finite Algorithm defines $\mathfrak{g}_0 = \mathfrak{g}^0 \oplus \mathbb{C}$. Identify $\mathbb{C}$ with $\mathbb{C}d_k$ where $d_k x = tx$ for $x \in \mathfrak{g}_1$. Use Lemma 7 to extend the definition of $(\ldots | \ldots)^0_\mathfrak{h}$ to all of $\mathfrak{h} = \mathfrak{h} + \mathbb{C}d_k$. Decree that $d_k$ is orthogonal to $x \in \mathfrak{g}$ if $x \not\in \mathfrak{h}$. Let $\nu : \mathfrak{h} \rightarrow \mathfrak{h}^*$ be the isomorphism determined by the nondegenerate form on $\mathfrak{h}$. Define $\Lambda_k := \nu(d_k)$, and $\alpha_k := \gamma_1 + (1/t_k)\Lambda_k = -\lambda_{-1} + (1/t_k)\Lambda_k$. The rest of the previous paragraph now applies without change.

We now have the elements necessary to define all the brackets on the graded pieces of $\mathfrak{g}$ as in §3. The remainder of the discussion goes towards verifying that the $\alpha_k$-extended $\mathcal{D}$ actually describes $\mathfrak{g}$ as a Lie algebra with those brackets.

### 6.1 Brackets

Let $[\cdot, \cdot] : \mathfrak{g}_{-1} \otimes \mathfrak{g}_1 \rightarrow \mathfrak{g}_0$ be given by

$$[u_{-1}, u] = -\sum_i (u_{-1}, X_i, u) Y_i + \kappa(u_{-1}, u) \tag{11}$$

where in the affine case, $\kappa = (-1/\alpha_k)K$ and in the finite case, $\kappa = (-1/t_k)d_k$.

Defining the rest of the brackets as in §3, we must show that $\mathcal{T}_1$ is in the kernel of the bracket defined on $\Lambda^2 V = \Lambda^2 \mathfrak{g}_1$ and, for $i > 2$, that $\mathcal{T}_i$ is in the kernel of the bracket defined on $\mathfrak{g}_1 \otimes \mathfrak{g}_i$. We verify that the bracket maps $\mathfrak{g}_1 \otimes \mathfrak{g}_i$ onto $\mathfrak{g}_{i+1}$ by producing a highest weight vector of $\mathfrak{g}_{i+1}$ in terms of the bracket. Operators help advance these arguments.

Let $\{X_i\}$ and $\{Y_i\}$ be Killing dual bases of $\mathfrak{g}$. Define $\Phi$ on $\Lambda^2 \mathfrak{g}_1$ by $\Phi(u \wedge v) = \sum_i X_i, v \wedge Y_i, u$.

**Proposition 7** In the affine case, $\Phi|_{\mathcal{T}_1} \equiv 0$. In the finite case, $\Phi|_{\mathcal{T}_1} \equiv -1/t_k$.

**Proof** We have $C.(u \wedge v) = 2c_1(u \wedge v) + 2\Phi(u \wedge v)$. The result follows by Proposition 3. □

We take $[u v] = \Phi(v \wedge u)$ in the affine case and in the finite case, $[u v] = \Phi(v \wedge u) + \frac{1}{t_k}v \wedge u$ (cf. Eqs. (3) and (4), §3.3.) Invoking Proposition 7 we get the following.

**Theorem 6** The bracket is identically zero on $\mathcal{T}_1$. □

**Lemma 35** A highest weight vector of $\mathfrak{g}_2$ has the form $[v^+] f_{\beta_1} f_{\beta_2} f_{\beta_3} v^+$, the $\beta_i$s as given in Proposition 2.
Proof In the affine case, we have
\[
[v^+ f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+] = -\sum_i X_i.v^+ \wedge Y_i.f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+ \\
= -(\lambda_1, \lambda_1 - \beta_1 - \beta_2 - \beta_3)v^+ \wedge f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+ + \sum v_\nu \wedge v_\mu \\
= v^+ \wedge f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+ + \sum v_\nu \wedge v_\mu
\]
where \(v_\nu\) and \(v_\mu\) are weight vectors in \(g_1\) with \(\nu < \lambda_1\) and \(\mu > \lambda_1 - \beta_1 - \beta_2 - \beta_3\). Note in particular that \(v\) is always of the form \(\lambda_1 - \alpha\) for a positive root \(\alpha\) and \(\mu = \lambda_1 - \beta_1 - \beta_2 - \beta_3 + \alpha\). Since \(v^+ \wedge f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+ \neq 0\), \([v^+ f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+] \neq 0\).

As the component of \([v^+ f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+]\) in \(\mathfrak{T}_1\) is zero, \([v^+ f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+]\) belongs to \(g_2\) and by weight considerations, the result follows.

In the finite case,
\[
[v^+ f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+] = -\sum_i X_i.v^+ \wedge Y_i.f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+ - \frac{1}{t_k} v^+ \wedge f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+ = \\
- \left(\lambda_1, \lambda_1 - \beta_1 - \beta_2 - \beta_3\right) + \frac{1}{t_k} v^+ \wedge f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+ + \sum v_\nu \wedge v_\mu = \\
\left(1 - \frac{1}{t_k}\right) v^+ \wedge f_{\beta_1} f_{\beta_2} f_{\beta_3}, v^+ + \sum v_\nu \wedge v_\mu.
\]

For reasons cited above, this is nonzero and in \(g_2\) when \(t_k \neq 1\). Even if \(t_k = 1\), though, it is clear that \(f_{\beta_1}, v^+ \wedge f_{\beta_2} f_{\beta_3}, v^+\), a multiple of one of the \(v_\nu \wedge v_\mu\) terms, is nonzero. The constant factor itself would be a nonzero multiple of \((\lambda_1 - \beta_2 - \beta_3, \beta_1) = 1\). Thus in these cases as well, the result holds. □

Lemma 36 For \(u_\alpha \in g_1\) and \(v, w \in g_1\), we have, in the affine case
\[
[u_\alpha [v w]] = \sum_i -(u_\alpha, X_i.v)Y_i.w + (u_\alpha, X_i.w)Y_i.v.
\]

In the finite case,
\[
[u_\alpha [v w]] = \sum_i -(u_\alpha, X_i.v)Y_i.w + (u_\alpha, X_i.w)Y_i.v - \frac{1}{t_k} (u_\alpha, v)w + \frac{1}{t_k} (u_\alpha, w)v.
\]

Proof By the Jacobi identity we have
\[
[u_\alpha [v w]] = [ [u_\alpha v w] + [v u_\alpha w] ] = [ [u_\alpha v]w] - [u_\alpha w]v.
\]
The result then follows application of Eqs. (3), (4) from §3.3, along with Eq. (11).

□

For \(t < j\), define \(\Psi\) on \(g_1 \otimes g_t\) by
\[
\Psi(u \otimes [u_1 \ldots u_t]) = \sum_i X_i.u \otimes Y_i.[u_1 \ldots u_t] + X_i.u_2 \otimes [u_1 Y_i.[u_3 \ldots u_t]] + \\
X_i.u_3 \otimes [u_1 u_2 Y_i.[u_4 \ldots u_t]] \ldots + X_i.u_{t-1} \otimes [u_1 \ldots u_{t-2} Y_i.u_t] \\
- X_i.u_t \otimes [u_1 \ldots u_{t-2} Y_i.u_{t-1}].
\]

36
Proposition 8 $\Psi$ is $\mathfrak{g}$-equivariant.

Proof We can view $\Psi$ as a sum of compositions of two kinds of mappings:
(1) permutations on multivectors in $\mathfrak{g}_1 \otimes \mathfrak{g}_t \subset \mathfrak{g}_1 \otimes \ldots \otimes \mathfrak{g}_1$; and (2) Casimir polarizations on multivectors, that is, mappings of the form

$$u \otimes u_1 \otimes \ldots \otimes u_t \mapsto \sum_{i} u \otimes \ldots \otimes u_{i-1} \otimes X_i u_i \otimes u_{i+1} \otimes \ldots \otimes u_{m-1} \otimes Y_i u_m \otimes u_{m+1} \otimes \ldots u_t.$$

This is a tedious but straightforward verification that one enacts starting with the observation that

$$\Psi(u_1 \otimes u_2) = \Phi(u_1 \wedge u_2) = \sum_i X_i u_1 \otimes Y_i u_2 - X_i u_2 \otimes Y_i u_1$$

can be realized as a composition of mappings

$$u_1 \otimes u_2 \mapsto u_1 \otimes u_2 - u_2 \otimes u_1$$

and

$$u_1 \otimes u_2 \mapsto \sum_{i} X_i u_1 \otimes Y_i u_2.$$

Both types of mappings are $k\mathfrak{g}$-equivariant. The first is the difference between the identity and the usual action of the symmetric group on $\mathfrak{g}_1 \otimes \mathfrak{g}_1$. The second is $C^2 - c_1$. Since $C$ is $\mathfrak{g}$-equivariant, its polarization is as well. □

It is convenient to think of $\Psi$ as a sum of three operators so define

$$\Psi_1(u \otimes [u_1 \ldots u_t]) = \sum_i X_i u \otimes Y_i [u_1 \ldots u_t],$$

$$\Psi_2(u \otimes [u_1 \ldots u_t]) = \sum_i X_i u_1 \otimes [u Y_i [u_2 \ldots u_t]],$$

$$\Psi_3 = \Psi - \Psi_1 - \Psi_2.$$

The argument we advanced to support the claim that $\Psi$ is $k\mathfrak{g}$-equivariant applies to give us $\Psi_1$, $\Psi_2$ and $\Psi_3$ $\mathfrak{g}$-equivariant as well.

Recall that $Y_{1,t}$ is the Cartan product of $\mathfrak{g}_1$ and $\mathfrak{g}_t$. Let $U_{1,t}$ be the submodule of $\mathfrak{g}_1 \otimes \mathfrak{g}_t$ with highest weights of the form $\lambda_1 + \lambda_t - \alpha, \alpha \in \Delta_+.$

We note the following.

Lemma 37 In the affine case, $\Psi_1|_{Y_{1,t}} \equiv 1$ and $\Psi_1|_{U_{1,t}} \equiv 1 - (\rho, \alpha)$. In the finite case, $\Psi_1|_{Y_{1,t}} \equiv 1 - \frac{t}{t_{\mathfrak{g}}} \rho$ and $\Psi_1|_{U_{1,t}} \equiv 1 - \frac{t}{t_{\mathfrak{g}}} - (\rho, \alpha)$.

Proof It is clear that $\Psi_1 = \frac{C - (c_1 + c_2)}{2}$. The result then follows Corollary 2. □
Let \( w^+ = \sum v_\nu \otimes v_\mu \) be a highest weight vector of \( \mathfrak{g}_t \), so that \( v_\nu \in \mathfrak{g}_1 \) and \( v_\mu \in \mathfrak{g}_{t-1} \). Take \( v_\nu \otimes v_\mu \) so that \( \nu \) is highest among the weights of \( \mathfrak{g}_1 \) associated to \( w^+ \).

Since \( e_\alpha.w^+ = 0 \) for all positive root vectors \( e_\alpha \in \mathfrak{g}_k \), it must be the case that \( (e_\alpha,v_\nu) \otimes v_\mu = 0 \) for all \( \alpha \) or that \( w^+ \) has a component of the form \( v_{\nu+\alpha} \otimes v_{\mu-\alpha} \). The latter contradicts our choice of \( \nu \) as maximal so \( e_\alpha.v_\nu \otimes v_\mu = 0 \) for all \( \alpha \in \Delta_+ \) implies \( w^+ \) has a component of the form \( v^+ \otimes v_\mu \).

**Proposition 9** If \( s \leq j \), a highest weight vector for \( \mathfrak{g}_s \) has the form

\[
\nu_s^+ = [v^+ f_\gamma_1 f_{\gamma_2} v^+ \cdots f_\nu f_{\nu_2} v^+ f_\beta_1 f_{\beta_2} f_\beta_3 v^+] 
\]

where \( \{\gamma_1, \gamma_2\}, \{\nu_1, \nu_2\}, \{\beta_1, \beta_2, \beta_3\} \) are sets of positive roots as in Proposition 2. Moreover, for \( v_\nu \in \mathfrak{g}_1 \), and \( v_\mu \in \mathfrak{g}_{t-1} \), \( [v_\nu v_\mu] = 0 \) when \( \nu + \mu > \lambda_s \). In particular, the bracket projects \( v_\nu \otimes v_\mu \) into \( \mathfrak{g}_s \).

We prove the result by induction. The case \( s = 2 \) was done in Lemma 35. The next sequence of lemmas follow the assumption that the proposition is true for all \( s \leq t \). At the end of those lemmas, we will have a proof of the proposition.

**Lemma 38** In the affine case, \( \Psi_2|_{Y_{1,t}} \equiv -1 \). In the finite case, \( \Psi_2|_{Y_{1,t}} \equiv -1 - \frac{t - 1}{t_k} \).

**Proof** A highest weight vector of \( Y_{1,t} \) has the form

\[
v^+ \otimes [v^+ f_\gamma_1 f_{\gamma_2} v^+ \cdots f_\nu f_{\nu_2} v^+ f_\beta_1 f_{\beta_2} f_\beta_3 v^+]\]

which we write \( v^+ \otimes [v^+ u] \). Note that \( u \in \mathfrak{g}_{t-1} \).

We have

\[
\Psi_2(v^+ \otimes [v^+ u]) = \sum_i X_i.v^+ \otimes [v^+ Y_i.u].
\]

Terms of (13) associated to \( X_i \) from positive root spaces are zero because \( X_i.v^+ = 0 \). Terms associated to \( Y_i \) from positive root spaces are also zero because then \( [v^+ Y_i.u] \) has a higher weight than the highest weight of \( \mathfrak{g}_t \). Thus the nonzero terms in (13) are contributed from the part of \( \{X_i\} \) and \( \{Y_i\} \) that comprise dual bases of \( \mathfrak{h} \). Invoking Corollary 2, we get, in the affine case,

\[
\Psi_2(v^+ \otimes [v^+ u]) = (\lambda_1, \lambda_t - \lambda_1)v^+ \otimes [v^+ u] = -v^+ \otimes [v^+ u].
\]

In the finite case, we have

\[
\Psi_2(v^+ \otimes [v^+ u]) = (\lambda_1, \lambda_t - \lambda_1)v^+ \otimes [v^+ u] = \left(-1 - \frac{t - 1}{t_k}\right)v^+ \otimes [v^+ u]. \Box
\]

**Lemma 39** In the affine case, \( \Psi|_{Y_{1,t}} \equiv 0 \).
Proof We have \((\Psi_1 + \Psi_2)|_{Y_{1,t}} \equiv 0\) in the affine case so it remains to show that \(\Psi_3|_{Y_{1,t}} \equiv 0\).

Let \(v^+_t = [v^+ f_{\gamma_1}f_{\gamma_2}v^+ \ldots f_{\nu_1}f_{\nu_2}v^+ f_{\beta_1}f_{\beta_2}f_{\beta_3}v^+]\) be a highest weight vector for \(g_t\). We have

\[
\Psi_3(v^+ \otimes v^+_t) = \sum_i X_i f_{\gamma_1}f_{\gamma_2}v^+ \otimes [v^+ v^+ Y_i[f_{\xi_1}f_{\xi_2}v^+ \ldots f_{\nu_1}f_{\nu_2}v^+ f_{\beta_1}f_{\beta_2}f_{\beta_3}v^+]] + \ldots
\]

\[\quad - X_i f_{\beta_1}f_{\beta_2}f_{\beta_3}v^+ \otimes [v^+ v^+ f_{\gamma_1}f_{\gamma_2}v^+ \ldots Y_i[f_{\nu_1}f_{\nu_2}v^+]]. \tag{14}\]

According to Proposition 2, \(\lambda_{t-1}\) has the form \((t-1)\lambda_1\) less \(2t-3\) positive roots. Consider that terms of \(\nu_{t-1}\) appearing in (14) with the form

\([v^+ v^+ f_{\gamma_1}f_{\gamma_2}v^+ \ldots Y_i[f_{\nu_1}f_{\nu_2}v^+ \ldots f_{\nu_1}f_{\nu_2}v^+ f_{\beta_1}f_{\beta_2}f_{\beta_3}v^+]\]

are brackets of \(t-1\) vectors from \(g_t\). The lowest weight that could be associated to such a term has the form \((t-1)\lambda_1\) less \(2(t-3) + 1 + 1 = 2t-4\) positive roots. Any such weight is strictly greater than \(\lambda_{t-1}\). By assumption, these brackets must all be zero, which proves the lemma. □

Comparing to Eq. (3) in §3.3, we see that the bracket is zero on \(Y_{1,t}\).

Lemma 40 In the finite case, \(\Psi_3|_{Y_{1,t}} \equiv - \frac{(2t-1)t}{t_k} \). □

Proof The proof that \(\Psi_3|_{Y_{1,t}} \equiv 0\) in the affine case applies here without modification. Lemmas 37 and 38 then apply to give us the result. □

Comparing to Eq. (4) in §3.3, we have

\[
[v^+ v^+ f_{\gamma_1}f_{\gamma_2}v^+ \ldots f_{\nu_1}f_{\nu_2}v^+ f_{\beta_1}f_{\beta_2}f_{\beta_3}v^+] := \]

\[\quad - \Psi(v^+ \otimes [v^+ f_{\gamma_1}f_{\gamma_2}v^+ \ldots f_{\nu_1}f_{\nu_2}v^+ f_{\beta_1}f_{\beta_2}f_{\beta_3}v^+]) - \]

\[\frac{2t-1}{t_k} [v^+ v^+ f_{\gamma_1}f_{\gamma_2}v^+ \ldots f_{\nu_1}f_{\nu_2}v^+ f_{\beta_1}f_{\beta_2}f_{\beta_3}v^+] = 0.\]

Note that, by assumption, all the other terms of Eq. (4) are zero: they involve brackets associated to weights that are too high, as in the proof of Lemma 39.

Lemma 41 In the affine case, \(\Psi_2|_{U_{1,t}} \equiv (\rho, \alpha) - 1\).

Proof We start by establishing that highest weight vectors in \(U_{1,t}\) can take a certain form. Choose \(\alpha \in \Delta_+\) with minimal height so that \(f_{\alpha}v^+\) and \(f_{\alpha}v^+_t\) are both nonzero, \(v^+_t = [v^+ u]\) a highest weight vector of \(g_t\). Let \(w_Y \in Y_{1,t}\) and \(w_U \in U_{1,t}\) satisfy

\[f_{\alpha}v^+ \otimes [v^+ w] = w_Y + w_U.\]

By weight considerations, \(w_U\) is a highest weight vector of \(U_{1,t}\), thus generates an irreducible submodule of \(U_{1,t}\). We assume that \(U_{1,t}\) is itself irreducible as doing so does not change our argument materially. We have

\[\Psi_1(f_{\alpha}v^+ \otimes v^+_t) = \sum_i X_i f_{\alpha}v^+ \otimes [Y_i v^+ w] + X_i f_{\alpha}v^+ \otimes [v^+ Y_i w].\]
Consider terms of the form
\[ X_i.f_\alpha.v^+ \otimes [Y_i.v^+ w]. \] (15)

Such an expression is nonzero only when \( X_i \) is from \( \mathfrak{k}_h \) or a positive root space: when \( X_i \) is from a negative root space, \( Y_i.v^+ = 0 \). If \( \{H_i\} \) and \( \{H'_i\} \) are dual bases of \( \mathfrak{k}_h \) we have
\[ \sum_i H_i.f_\alpha.v^+ \otimes [H'_i,v^+ w] = (\lambda_1, \lambda_1 - \alpha) f_\alpha.v^+ \otimes [v^+ w] = f_\alpha.v^+ \otimes [v^+ w]. \]

Recall next that there is only one simple root \( \alpha_i \) (per simple component of \( \mathfrak{k}_g \)) that satisfies \( f_\alpha_i.v^+ \neq 0 \). Moreover, the root \( \alpha \) appearing in (15) has minimal height with \( f_\alpha.v^+ \) and \( f_\alpha,[v^+ w] \) both nonzero. In particular, \( \alpha_i \) must be in the support of \( \alpha \), but with multiplicity one.

If \( X_i \) is from a positive root space, \( X_i.f_\alpha.v^+ \) is zero when the positive root associated to \( X_i \) has \( \alpha_i \) in support, unless \( X_i \) is associated to \( \alpha_i \) itself. We lose no generality in assuming \( f_\alpha \) is among the \( Y_i \) basis vectors. The nonzero terms in (15) then reduce to
\[ f_\alpha.v^+ \otimes [v^+ w] + v^+ \otimes [f_\alpha.v^+ w]. \]

This gives us
\[ \Psi_1(f_\alpha.v^+ \otimes [v^+ w]) = f_\alpha.v^+ \otimes [v^+ w] + v^+ \otimes [f_\alpha.v^+ w] + \]
\[ \sum_i X_i.f_\alpha.v^+ \otimes [v^+ Y_i.w] = (1 - (\rho, \alpha)) w_U + w_Y. \]

Then
\[ -(\rho, \alpha) w_U = v^+ \otimes [f_\alpha.v^+ w] + \sum_i X_i.v^+ \otimes [f_\alpha.v^+ Y_i.w]. \]

If we take
\[ v^+ \otimes [f_\alpha.v^+ w] = v_U + v_Y, \]
where \( v_U \in U_{1,t} \) and \( v_Y \in Y_{1,t} \), we can apply \( \Psi_1 \) to show that
\[ -(\rho, \alpha) v_U = f_\alpha.v^+ \otimes [v^+ w] + \sum_i X_i.v^+ \otimes [f_\alpha.v^+ Y_i.w]. \]

Let \( \kappa \) be the scalar value of \( \Psi_2 \) on \( U_{1,t} \). We have
\[ \Psi_2(w_U + w_Y) = \kappa w_U - w_Y = \sum_i X_i.v^+ \otimes [f_\alpha.v^+ Y_i.w] \]
and
\[ \Psi_2(v_U + v_Y) = \kappa v_U - v_Y = \sum_i X_i.f_\alpha.v^+ \otimes [v^+ Y_i.w]. \]
Then $-\langle \rho, \alpha \rangle v_U = (\kappa + 1)w_U$ and $-\langle \rho, \alpha \rangle w_U = (\kappa + 1)v_U$ imply that $\kappa + 1 = \pm(\rho, \alpha)$, or, equivalently, that $w_U = \mp v_U$. Suppose $w_U = v_U$. We then have

$$z = w_U + w_Y - (v_U + v_Y) = f_{\alpha^+}v^+ \otimes [v^+ w] - v^+ \otimes [f_{\alpha}v^+ w] \in Y_{1,t}.$$ 

Note that $e_i [v^+ w] = [v^+ e_i w] = 0$ for any positive root vector $e_i \in \mathfrak{h}$, by assumption. Suppose $\alpha$ is simple. Then $e_i z = 0$ for all $e_i$ so $z$, a weight vector with weight $\lambda_1 + \lambda_t - \alpha$, is a highest weight vector of $Y_{1,t}$. That is impossible since $Y_{1,t}$ is irreducible with highest weight $\lambda_1 + \lambda_t$. If $\alpha$ is not simple, choose a positive root vector $e_i \in \mathfrak{h}$ so that $e_i f_{\alpha}v^+ \neq 0$. Let $[e_i f_{\alpha}] = f_{\gamma}$ and note that the height of $\gamma$ is less than the height of $\alpha$. We have

$$e_i z = f_{\gamma}v^+ \otimes [v^+ w] - v^+ \otimes [f_{\gamma}v^+ w] \in Y_{1,t}.$$ 

Repeating the procedure as necessary, we get a highest weight vector in $Y_{1,t}$ associated to a weight strictly less than $\lambda_1 + \lambda_t$, a contradiction. Our conclusion is that $w_U = -v_Y$, and $\kappa = \rho, \alpha - 1$. □

**Lemma 42** In the finite case, $\Psi_{2|U_{1,t}} \equiv (\rho, \alpha) - \frac{t-1}{t_k} - 1$.

**Proof** As above, we take $f_{\alpha}v^+ \otimes [v^+ w] = w_U + w_Y$. On the one hand, we have

$$\Psi_1(f_{\alpha}v^+ \otimes [v^+ w]) = \sum_i X_i f_{\alpha}v^+ \otimes Y_i[v^+ w] = (\lambda_1 - \alpha, \lambda_1) f_{\alpha}v^+ \otimes [v^+ w] + v^+ \otimes [f_{\alpha}v^+ w] + \sum_i X_i f_{\alpha}v^+ \otimes [v^+ Y_i w] =$$

$$(1 - \frac{1}{t_k}) f_{\alpha}v^+ \otimes [v^+ w] + v^+ \otimes [f_{\alpha}v^+ w] + \sum_i X_i f_{\alpha}v^+ \otimes [v^+ Y_i w].$$

On the other hand, we have

$$\Psi_1(f_{\alpha}v^+\otimes[ v^+ w]) = \Psi_1(w_U) + \Psi_1(w_Y) = \left(1 - \frac{t}{t_k} - (\rho, \alpha)\right) w_U + \left(1 - \frac{t}{t_k}\right) w_Y.$$ 

Subtract

$$\left(1 - \frac{t}{t_k}\right) f_{\alpha}v^+ \otimes [v^+ w] = \left(1 - \frac{t}{t_k}\right) (w_U + w_Y)$$

from both expressions to get

$$-\langle \rho, \alpha \rangle w_U = \left(\frac{t-1}{t_k}\right) f_{\alpha}v^+ \otimes [v^+ w] + v^+ \otimes [f_{\alpha}v^+ w] + \sum_i X_i f_{\alpha}v^+ \otimes [v^+ Y_i w].$$

Applying the same calculations to $w_U + w_Y = v^+ \otimes [f_{\alpha}v^+ w]$ we get

$$-\langle \rho, \alpha \rangle v_U = \left(\frac{t-1}{t_k}\right) v^+ \otimes [f_{\alpha}v^+ w] + f_{\alpha}v^+ \otimes [v^+ w] + \sum_i X_i v^+ \otimes [f_{\alpha}v^+ Y_i w].$$
Let $\tau$ be the scalar value of $\Psi_2$ on $U_{1,t}$, assuming $U_{1,t}$ is irreducible. We have

$$\Psi_2(w_U + w_Y) = \tau w_U + \left(-1 - \frac{t-1}{t_k}\right) w_Y = \sum_i X_i.v^+ \otimes [f_{a,v^+} Y_i.w],$$

and

$$\Psi_2(v_U + v_Y) = \tau v_U + \left(-1 - \frac{t-1}{t_k}\right) v_Y = \sum_i X_i.f_{a,v^+} \otimes [v^+ Y_i.w].$$

Rewrite (16) to get

$$-(\rho, \alpha)w_U = \frac{t-1}{t_k}(w_U + w_Y) + (v_U + v_Y) + \left(-1 - \frac{t-1}{t_k}\right)v_Y + \tau v_U.$$

This gives us

$$\left(-(\rho, \alpha) + \frac{1-t}{t_k}\right) w_U = (1 + \tau)v_U + \frac{t-1}{t_k}(w_Y - v_Y).$$

Since $t > 1$, $w_Y - v_Y = 0$ giving us

$$\left(-(\rho, \alpha) + \frac{1-t}{t_k}\right) w_U = (1 + \tau)v_U.$$

The same trick applied to (17) gives us

$$\left(-(\rho, \alpha) + \frac{1-t}{t_k}\right) v_U = (1 + \tau)w_U,$$

which implies

$$\left(-(\rho, \alpha) + \frac{1-t}{t_k}\right)^2 = (1 + \tau)^2.$$

Then \left(-(\rho, \alpha) + \frac{1-t}{t_k}\right) = \pm(1 + \tau), or equivalently, $w_U = \pm v_U$. We established that $w_Y = v_Y$, though, so if $w_U = v_U$, we have $f_{a,v^+} \otimes [v^+ w] = v^+ \otimes [f_{a,v^+} w]$, which is absurd. We conclude that

$$\left(-(\rho, \alpha) + \frac{1-t}{t_k}\right) = -(1 + \tau)$$

giving us $\tau = (\rho, \alpha) + \frac{t-1}{t_k} - 1$, as desired. □

**Lemma 43** In both the affine and finite cases, $\Psi_3|_{U_{1,t}} \equiv 0$.

**Proof** We do the proof for the affine case; there is no significant difference in the finite case.
We have
\[ f_\alpha.v^+ \otimes [v^+ w] = w_Y + w_U = w_Y - \frac{1}{(\rho, \alpha)} v^+ \otimes [f_\alpha.v^+ w] - \frac{1}{(\rho, \alpha)} \sum_i X_i.f_\alpha.v^+ \otimes [v^+ Y_i.w]. \] (18)
This is for a positive root \( \alpha \) so that \( \lambda_1 + \lambda_\ell - \alpha \) is a highest weight of \( U_{1, t} \).
We know that \( \Psi_3 \) is homothetic on an irreducible component of \( U_{1, t} \). As before, we lose no generality assuming that \( U_{1, t} \) is irreducible. Say that \( \Psi_3|_{U_{1, t}} \equiv \tau \). As per Proposition 9, take
\[ [v^+ w] = [v^+ f_\gamma_1.f_\gamma_2.v^+ \ldots f_\nu_1.f_\nu_2.v^+ f_{\beta_1}.f_{\beta_2}.f_{\beta_3}.v^+]. \]
We have
\[ \Psi_3(f_\alpha.v^+ \otimes [v^+ w]) = \sum_i X_i.f_\gamma_1.f_\gamma_2.v^+ \otimes [f_\alpha.v^+ v^+ Y_i.[f_{\xi_1}.f_{\xi_2}.v^+ \ldots f_\nu_1.f_\nu_2.v^+ f_{\beta_1}.f_{\beta_2}.f_{\beta_3}.v^+]] + \ldots \]
\[ + X_i.f_\beta_1.f_\beta_2.f_\gamma_2.v^+ \otimes [f_\alpha.v^+ v^+ f_\gamma_1.f_\gamma_2.v^+ \ldots f_{\xi_1}.f_{\xi_2}.v^+ Y_i.f_{\beta_1}.f_{\beta_2}.f_{\beta_3}.v^+] \]
\[ - X_i.f_\beta_1.f_\beta_2.f_\beta_3.v^+ \otimes [f_\alpha.v^+ v^+ f_{\gamma_1}.f_{\gamma_2}.v^+ \ldots f_{\xi_1}.f_{\xi_2}.v^+ Y_i.f_{\gamma_1}.f_{\gamma_2}.v^+] \]
\[ = -\frac{\tau}{(\rho, \alpha)} v^+ \otimes [f_\alpha.v^+ w] - \frac{\tau}{(\rho, \alpha)} \sum_i X_i.f_\alpha.v^+ \otimes [v^+ Y_i.w]. \] (19)
If the expression on the right side of the last equal sign in Eq. (19) does not include a nonzero multiple component of \( v^+ \otimes [f_\alpha.v^+ w] \), then some \( X_i.f_\alpha.v^+ \otimes [v^+ Y_i.w] = -v^+ \otimes [f_\alpha.v^+ w] \). Without loss of generality, we can assume that \( v^+ \otimes [v^+ f_\alpha.w] = -v^+ \otimes [f_\alpha.v^+ w] \). In this case,
\[ f_\alpha.(v^+ \otimes [v^+ w]) = f_\alpha.v^+ \otimes [v^+ w] \in Y_{1, t}. \]
Comparing to (18), we are forced to conclude that
\[ v^+ \otimes [f_\alpha.v^+ w] + \sum_i X_i.f_\alpha.v^+ \otimes [v^+ Y_i.w] = 0, \]
which is impossible unless \( U_{1, t} \) is itself zero.
If \( U_{1, t} \) is not zero, the expression on the right side of the equal sign in Eq. (19) includes a nonzero multiple of \( v^+ \otimes [f_\alpha.v^+ w] \) but the left side does not, as terms on the left side are all of the form \( v_\nu \otimes v_\mu \), \( v_\nu \in \mathfrak{g}_1 \) with \( \nu \) no greater than \( \lambda_1 \) less one positive root of \( \mathfrak{k} \). We conclude that \( \tau = 0 \). □

**Lemma 44** In the finite case, \( \Psi|_{U_{1, t}} \equiv \frac{1}{\tau_\mathfrak{k}} \).

**Proof** The proof follows the lemmas. □

If we define the bracket in the affine case to be \( \Psi \) and in the finite case to be \( \Psi + \frac{1}{\tau_\mathfrak{k}} \), then it is identically zero on \( \mathcal{T}_t \). Proposition 9 now follows by a weight argument applied to terms of the form (12). This proves that the algorithms actually produce Lie algebras that enjoy the structure determined by \( \mathcal{D} \) as extended by \( \alpha_\mathfrak{k} \), which completes the proofs of Theorems 1 and 2.
References

[1] Bourbaki, N., “Groupes et algèbres de Lie”, Hermann, Paris, 1968.

[2] Dillon, M. I., Weight strings in nonstandard representations of Kac-Moody algebras, Trans. Amer. Math. Soc. 320 (1990), 161 – 169.

[3] Gorbatsevich, V. V., A. L. Onishchik, and E. B. Vinberg, Lie groups and Lie algebras III: Structure of Lie groups and Lie algebras, Encyclopaedia of Mathematical Sciences Volume 41, Springer-Verlag, Berlin, 1994.

[4] Humphreys, J. E., “Introduction to Lie algebras and Representation Theory,” Springer-Verlag, New York, 1972.

[5] Kac, V. G., “Infinite dimensional Lie algebras, Third Edition,” Cambridge University Press, 1990.

[6] Kostant, B., A cubic Dirac operator and the emergence of Euler number multiplets of representations for equal rank subgroups, Duke Math. J. 100 (1999), 447 – 501.

[7] Landsberg, J. M., and L. Manivel, Construction and classification of complex simple Lie algebras via projective geometry, Selecta Mathematica 8 (2002), 137 – 159.

[8] Magyar, Peter, Littelmann paths for the basic representation of an affine Lie algebra, preprint, 2003.

[9] van Leeuwen, Marc, LiE: A Computer algebra package for Lie group computations, http://young.sp2mi.univ-poitiers.fr/ marc/LiE/

Meighan I. Dillon
Mathematics
Southern Polytechnic State University
1100 S. Marietta Pkwy, Marietta, GA 30060
mdillon@spsu.edu