INTRINSIC CONSTRUCTION OF INVARIANT FUNCTIONS ON SIMPLE LIE ALGEBRAS

ZHAOHU NIE

ABSTRACT. An algorithm for constructing primitive adjoint-invariant functions on a complex simple Lie algebra is presented. The construction is intrinsic in the sense that it does not resort to any representation. A primitive invariant function on the whole Lie algebra is obtained by lifting a coordinate function on a Kostant slice of the Lie algebra. Such an intrinsic construction of invariant functions is most useful for the bigger exceptional Lie algebras such as the $E$’s. The Maple implementation of this algorithm is outlined at the end and will be applied to these exceptional Lie algebras in a future work.

1. INTRODUCTION

Let $\mathfrak{g}$ be a complex simple Lie algebra of rank $l$ with adjoint group $G$. We recall that $G$ acts on $\mathfrak{g}$ by the adjoint action, and therefore on the algebra $\mathcal{P}(\mathfrak{g})$ of polynomials on $\mathfrak{g}$ by its contragredient, that is,

$$ (g \cdot P)(x) = P(\text{Ad}_g^{-1}x), \quad g \in G, \quad P \in \mathcal{P}(\mathfrak{g}), \quad x \in \mathfrak{g}. $$

Let

$$ I(\mathfrak{g}) = \mathcal{P}(\mathfrak{g})^G $$

be the algebra of polynomials on $\mathfrak{g}$ invariant under the above action of $G$. A well-known theorem of Chevalley [Che55] asserts that $I(\mathfrak{g})$ is polynomial algebra on $l$ homogeneous polynomials $I_1, \ldots, I_l$, that is,

$$ I(\mathfrak{g}) = \mathbb{C}[I_1, \ldots, I_l]. $$

We will refer to the $I_j$’s as primitive invariant functions on $\mathfrak{g}$. Write the degrees

$$ \deg I_j = d_j, \quad j = 1, \ldots, l. $$

We will assume that the $I_j$’s are ordered in the sense that

$$ d_1 \leq d_2 \leq \cdots \leq d_l. $$

The numbers

$$ m_j = d_j - 1, \quad j = 1, \ldots, l, $$

are called the exponents of $\mathfrak{g}$.

Although the choice of the $I_j$’s is not unique, the degrees $d_j$ and hence the exponents $m_j$ are intrinsic to $\mathfrak{g}$ which constitute important invariants (see [Che52]).

Our main objective in this paper is to give an algorithm to explicitly and intrinsically construct a set of primitive invariant functions. We note that our invariant
functions are defined on the whole Lie algebra \( \mathfrak{g} \). We also comment that our construction is uniform, explains the pattern for the exponents, and does not resort to any representation. The author has implemented his algorithm on Maple.

The traditional way of obtaining such invariant functions on \( \mathfrak{g} \) is extrinsic by employing a faithful representation. Usually the first fundamental representation of the Lie algebra is used because of its small dimension. The representation expresses a general Lie algebra element as a matrix, whose dimension is the same as that of the representation space, involving its coordinates. The sum of the principal minors of this matrix with dimension equal to a degree \( d_j \) \((1.2)\) of the Lie algebra is one such sought-after primitive invariant function. To this author, there are several drawbacks to this approach. First, this construction uses the a priori information of the degrees \( d_j \) of \( \mathfrak{g} \) without being able to provide any deeper reason, and in the case of \( D_{2n} = \mathfrak{so}(4n) \) where the degree \( 2n \) has multiplicity 2, a special formula is needed for the Pfaffian. Furthermore for bigger exceptional Lie algebras, their representations are hard to be made explicit, and the enormous cardinality of the principal minors of a big matrix prevents this method from being efficient. In particular, explicit forms of invariant functions on \( E_8 \) are only known up to the second one of degree 8 \[CP07\].

In view of the above, an intrinsic and uniform method is clearly desirable. Intuitively speaking, our algorithm uses the restriction of the adjoint representation on a principal sl2 subalgebra in \( \mathfrak{g} \), and we gain independence from other representations and furthermore computational efficiency in this way. Together with his collaborator, the author plans to apply his Maple implementation of this algorithm to other invariant functions on the \( E \)’s, with the goal of at least getting some interesting restrictions of the invariant functions, that is, carrying out step (i) in Theorem 1.11. The restricted functions can still be very interesting and useful for the exceptional Lie algebras, since for example they are the functions most relevant to integrable systems such as Toda lattices \[Kos79\] and generic Toda flows \[GS99\].

The foundation for our construction is Kostant’s profound studies \[Kos59,Kos63,Kos78\] on invariant functions, which we now introduce. Fix a Cartan subalgebra \( \mathfrak{h} \subset \mathfrak{g} \). Let \( \Delta \) be the corresponding root system, \( \Delta_{\pm} \) a choice of positive/negative roots, and \( \pi = \{ \alpha_1, \cdots, \alpha_l \} \) the positive simple roots. Let \( \mathfrak{g} = \mathfrak{h} \oplus \bigoplus_{\alpha \in \Delta} \mathfrak{g}_\alpha \) be the root space decomposition, with \( \mathfrak{g}_\alpha \) generated by a root vector \( e_\alpha \). For \( \alpha \in \Delta_+ \), let \( H_\alpha = \{ e_\alpha, e_{-\alpha} \} \), and we require \( \alpha(H_\alpha) = 2 \) for the choices of root vectors. For \( 1 \leq i \leq l \), the \( H_{\alpha_i} \) form a basis of \( \mathfrak{h} \).

The height (or the order) \( o(\alpha) \) of a root \( \alpha \in \Delta \) is defined as

\[
(1.4) \quad o(\alpha) = \sum_{i=1}^{l} n_{i}, \quad \text{if} \quad \alpha = \sum_{i=1}^{l} n_{i} \alpha_{i}.
\]

This also induces a height gradation

\[
(1.5) \quad \mathfrak{g} \cong \bigoplus_{k} \mathfrak{g}_k, \quad \mathfrak{g}_k = \bigoplus_{o(\alpha) = k} \mathfrak{g}_\alpha \quad \text{and} \quad \mathfrak{g}_0 = \mathfrak{h}.
\]

For \( x \in \mathfrak{g}_k \), we write \( o(x) = k \) by abusing the notation and call \( k \) the height of \( x \). Let \( \mathfrak{n} = \bigoplus_{\alpha \in \Delta_+} \mathfrak{g}_\alpha = \bigoplus_{k > 0} \mathfrak{g}_k \) be the maximal nilpotent subalgebra of \( \mathfrak{g} \), \( \mathfrak{b} = \mathfrak{h} \oplus \mathfrak{n} \) the Borel subalgebra, and \( N \) the unipotent subgroup of \( G \) corresponding to \( \mathfrak{n} \).
Define

\[(1.6) \quad \epsilon = \sum_{i=1}^{l} e_{-\alpha_i}.\]

Let \(s\) be a complement of \([\epsilon, \mathfrak{g}]\) in \(\mathfrak{g}\), that is,

\[(1.7) \quad \mathfrak{g} \cong s \oplus [\epsilon, \mathfrak{g}].\]

Then by [Kos59, Kos63], \(s\) is a Kostant slice, and \(\dim(s) = l\) is equal to the rank. We call \(s\) a Kostant slice, and let \(\{s_j\}_{j=1}^{l}\) be a homogeneous basis of \(s\) with respect to the height gradation (1.5).

The following theorem summarizes several important results of Kostant on invariant functions.

**Theorem 1.8** (Kostant [Kos59, Kos63, Kos78]). The heights of the \(s_j\) are correspondingly the exponents of the Lie algebra \(\mathfrak{g}\). That is, if we order the \(\{s_j\}\) so that \(o(s_1) \leq o(s_2) \leq \cdots \leq o(s_l)\), then

\[(1.9) \quad o(s_j) = m_j, \quad 1 \leq j \leq l.\]

Furthermore, there is a sequence of isomorphisms through restrictions

\[(1.10) \quad I(\mathfrak{g}) \xrightarrow{r_1} \mathcal{P}(\epsilon + b)^N \xrightarrow{r_2} \mathcal{P}(\epsilon + s),\]

where \(\mathcal{P}(\epsilon + b)^N\) is the algebra of polynomials on \(\epsilon + b\) invariant under the \(N\) action, and \(\mathcal{P}(\epsilon + s)\) is the algebra of all polynomials on \(\epsilon + s\).

The following is our main result.

**Theorem 1.11.**

(i) There is an explicit algorithm for constructing the inverse to \(r_2\) in (1.10):

\[r_2^{-1} : \mathcal{P}(\epsilon + s) \to \mathcal{P}(\epsilon + b)^N.\]

More precisely, let the \(\xi_j\) be the coordinates of a general point

\[\epsilon + \sum_{j=1}^{l} \xi_j s_j \in \epsilon + s.\]

Then there is an algorithm for constructing \(l\) primitive invariant function \(I_j\) defined on \(\epsilon + b\) of degree \(d_j\) such that

\[(1.12) \quad I_j(\epsilon + \sum_{i=1}^{l} \xi_i s_i) = \xi_j, \quad 1 \leq j \leq l.\]

(ii) Furthermore, there is an explicit algorithm for constructing the inverse to \(r_1\) in (1.10) such that the invariant functions are defined on the whole \(\mathfrak{g}\).

We present our basic setup and computation techniques in Section 2. In Section 3, we present our algorithms in the proofs of the two parts of our main Theorem 1.11 together with several propositions. In part (i), our algorithm for constructing \(r_2^{-1}\) lifts the values of the invariant functions from the slice \(\epsilon + s\) to \(\epsilon + b\) and then to the Cartan subalgebra \(\mathfrak{h}\). In Proposition 3.18, we prove the Weyl invariance of the resulted function on \(\mathfrak{h}\). In part (ii), we present the similar algorithm for constructing \(r_1^{-1}\) to define the invariant functions on \(\mathfrak{g}\). The complexity of this step is much bigger than the previous step, and to a large extent accounts for the...
difficulty in getting the invariant functions on the whole Lie algebra. In Section 4, we outline how the author has implemented the algorithm on the software Maple.

Because of the special roles played by exceptional Lie algebras in mathematics and physics, we expect our algorithm and the explicit invariant functions it produces to have applications in a range of areas such as integrable systems and higher Casimir operators.

Acknowledgment. The author thanks Ian Anderson for help with Maple and Luen-Chau Li for interest in this work.

2. Setup and Computation Techniques

Our later calculations rely on the following setup of Kostant [Kos63] in an essential way. We have also extensively used such a setup and developed some techniques for computation in [LN11]. This paper is a further application of such techniques.

To begin with, we recall that the polynomial algebra $\mathfrak{P}(\mathfrak{g})$ can be identified with the symmetric algebra $\mathfrak{S}(\mathfrak{g}^*)$ on $\mathfrak{g}^*$, the dual of $\mathfrak{g}$. On the other hand, we can associate to each $x \in \mathfrak{g}$ a differential operator $\partial_x$ on $\mathfrak{g}$, defined by

$$ (\partial_x f)(y) = \frac{d}{dt} \bigg|_{t=0} f(y + tx), \quad f \in C^\infty(\mathfrak{g}). $$

In this way we have a linear map $x \mapsto \partial_x$ which can be extended to an isomorphism $\partial: S^* \to \mathfrak{g}^*$ given by

$$ \langle \partial, f \rangle = (\partial f)(0), \quad \partial \in S^*, \quad f \in S, $$

where $(\partial f)(0)$ denotes the value of the function $\partial f$ at $0 \in \mathfrak{g}$. It is clear that both $S^*$ and $S$ are graded from the tensor structure:

$$ S^* = \bigoplus_{k=0}^{\infty} S^k, \quad S = \bigoplus_{k=0}^{\infty} S^k, $$

and $S^j$ pairs nontrivially only with $S^j$.

If $f \in S^k$ and $x \in \mathfrak{g}$, it follows from the Taylor expansion that

$$ \langle \partial x \partial^k, f \rangle = f(x). $$

It is clear that the adjoint action of $G$ on $\mathfrak{g}$ can be naturally extended to an action of $G$ on $S^*$. On the other hand, $S$ is a $G$-module as its contragredient by (1.1). (We denote the actions of $G$ and later of $\mathfrak{g}$ by a dot.) We have

$$ \langle g \cdot \partial, g \cdot f \rangle = \langle \partial, f \rangle, \quad \forall g \in G, \partial \in S^*, \quad f \in S. $$

By differentiation, $S$ and $S^*$ become $\mathfrak{g}$-modules and the actions of $\mathfrak{g}$ on both spaces are by derivations. Therefore we have the following properties:

$$ [x, y] \cdot \partial = x \cdot (y \cdot \partial) - y \cdot (x \cdot \partial), \quad x, y \in \mathfrak{g}, \partial \in S^*, \quad \text{Lie alg hom} $$

$$ x \cdot \partial_y = \partial_{[x,y]}, \quad x, y \in \mathfrak{g}, \quad \text{adj action} $$

$$ x \cdot (\partial \delta) = (x \cdot \partial)\delta + \partial(x \cdot \delta), \quad x \in \mathfrak{g}, \partial, \delta \in S^*, \quad \text{deriviation} $$

$$ x \cdot \partial^n = n\partial^{n-1}(x \cdot \partial), \quad \text{power rule} $$

$$ f \in I(\mathfrak{g}) \implies x \cdot f = 0, \quad \forall x \in \mathfrak{g}, \quad \text{inv property} $$
Since the pairing between $S_*$ and $S$ obeys (2.4), it follows from derivation that the $g$-actions satisfy
\[
\langle x \cdot \partial, f \rangle + \langle \partial, x \cdot f \rangle = 0, \quad \forall x \in g, \partial \in S_*, \ f \in S.
\]
This and (2.9) imply that
\[
(2.10) \quad \langle x \cdot \partial, f \rangle = 0, \quad \forall f \in I(g), \ x \in g.
\]

There is a grading element $x_0 \in h$ defined by the conditions that
\[
(2.11) \quad \alpha_i(x_0) = 1, \quad \forall 1 \leq i \leq l.
\]
By (2.11), $\alpha(x_0) = o(\alpha)$, and $[x_0, e_\alpha] = o(\alpha)e_\alpha$. Thus the graded subspaces from (1.5) are $g_k = \{ x \in g \mid [x_0, x] = kx \}$. This motivate the following definition of the weight structure of \([\text{Kos59}]\) on $S_*$. For each $k \in \mathbb{Z}$,
\[
(2.12) \quad S_k = \{ \partial \in S_* \mid x_0 \cdot \partial = k\partial \}.
\]

Applying (2.10) to $x_0 \in h$ gives us the first vanishing result of Kostant \([\text{Kos63}]\).

If $\partial \in S_k$ for $k \neq 0$, then by (2.12), $\partial = \frac{1}{k} x_0 \cdot \partial$. Then in view of (2.10), we have
\[
(2.13) \quad \partial \in S_k \text{ for } k \neq 0 \Rightarrow \langle \partial, f \rangle = 0 \text{ for all } f \in I(g).
\]

Applying (2.10) to a general $x \in h$ gives us the following refined vanishing.

**Lemma 2.14** ([\text{Kos78}], [\text{LN11}, Lemma 3.1]). For all $f \in I(g)$, $p \in h$,
\[
\left\langle \partial_p^n \prod_{\alpha \in \Delta} \partial_{e_\alpha}^{m_\alpha}, f \right\rangle = 0
\]
unless $\sum_{\alpha \in \Delta} m_\alpha \alpha = 0$.

In our later constructions, we will further exploit (2.10) by applying it to other elements in $g$. The most convenient formulation for us is the following “integration by parts” formula to “move things around.”

**Lemma 2.15** ([\text{LN11}, Lemma 6.2]). Let $f \in I(g)$, then for all $x, y \in g$, $\partial \in S_*$, and $m \geq 0$, we have
\[
(2.16) \quad \langle \partial_x^m \partial_{[x,y]} \partial, f \rangle = \frac{1}{m+1} \langle \partial_x^{m+1}(y \cdot \partial), f \rangle.
\]

For the reader’s convenience, we repeat the brief proof.

**Proof.** By using (2.6), (2.8), (2.7) and (2.10), we find that
\[
\langle \partial_x^m \partial_{[x,y]} \partial, f \rangle = -\langle \partial_x^m (y \cdot \partial_x) \partial, f \rangle
\]
\[
= -\frac{1}{m+1} \langle (y \cdot \partial_x^{m+1}) \partial, f \rangle
\]
\[
= -\frac{1}{m+1} \langle y \cdot (\partial_x^{m+1} \partial), f \rangle + \frac{1}{m+1} \langle \partial_x^{m+1}(y \cdot \partial), f \rangle
\]
\[
= \frac{1}{m+1} \langle \partial_x^{m+1}(y \cdot \partial), f \rangle.
\]

$\square$
3. Constructive proof of the main theorem

In this section, we present our algorithms in the forms of proofs to the two parts of our main Theorem 1.11 and we present several supporting propositions. At the end of this section, we show how to assemble the information we obtain to get the concrete form of an invariant function.

First we present a different basis of \( g \), which is crucial to our inductive procedures later. Define

\[
s_j^k = (\text{ad}_\epsilon)^k s_j, \quad 0 \leq k \leq 2m_j, \quad 1 \leq j \leq l.
\]

Then \( s_j^k = [\epsilon, s_j^{k-1}] \) for \( k \geq 1 \).

**Lemma 3.2.** \( \{s_j^k\}_{0 \leq k \leq 2m_j} \) is a basis of \( g \).

**Proof.** This follows from (1.7) and Kostant’s work in [Kos59]. □

The height of \( s_j^k \) is \( m_j - k \) by (1.9) and (1.6), and therefore

\[
U := \{s_j^k\}_{0 \leq k \leq m_j-1} \text{ is a basis of } n = \bigoplus_{k>0} \mathfrak{g}_k.
\]

We will denote a general element of \( U \) by \( u \). Note that for such \( u \)'s, either \( u \in s \) or \( u \in [\epsilon, g] \) in the decomposition (1.7). (Actually the preimage of such a \( u \in \text{im}(\text{ad}_\epsilon) \) is unique since \( \text{ad}_\epsilon \) has no kernel in \( b \) [Kos59].)

Similarly we have that

\[
W := \{s_j^k\}_{m_j+2 \leq k \leq 2m_j} \text{ is a basis of } \bigoplus_{k \leq -2} \mathfrak{g}_k.
\]

We will denote a general element of \( W \) by \( w \). Note that all such \( w \)'s belong to \( [\epsilon, g] \) in (1.7).

By the special nature of \( \mathfrak{g}_0 = \mathfrak{h} \) and \( \mathfrak{g}_{-1} \), this author will use their natural basis \( \{H_\alpha\}_{1 \leq j \leq l} \) and \( \{\epsilon - \alpha\}_{1 \leq j \leq l} \) in the algorithms.

**Proof of Theorem 1.11 (i).** Since we want to construct invariant functions, we will enforce the invariance property (2.10) or more explicitly Lemma 2.15. In this part, we will inductively show that this rule and the condition (1.12) determine the invariant function \( I_j \) on \( \epsilon + b \). Then we prove that the constructed function when restricted on \( \mathfrak{h} \) is invariant under the Weyl group in Propositions 3.18. Proposition 3.20 quickly shows that such functions are algebraically independent.

In this proof, we will work with a fixed \( I_j \), and we often write \( I \) for short. Also \( d = d_j \) and \( m = m_j \).

Using (2.3) and the multinomial theorem, to describe \( I \in \mathcal{P}(\epsilon + b) \), we need to determine all the following polynomials

\[
\langle \partial_{u(1)} \partial_{u(2)} \cdots \partial_{u(c)} \partial_{p}^a p^b, I \rangle
\]

of degree \( a \) in \( p \in \mathfrak{h} \). Here \( U = (u(1), \cdots, u(c)) \) is a sequence with each \( u(i) \) (possibly repeating) from \( U \) in (3.3). The expression (3.5) is nonzero only if

\[
a + b + c = d,
\]

by (2.2) and (2.13). (This is related to the \( x_0 \)-grading by Kahzdan in [Kos78].)
We run increasing induction on $a$ and decreasing induction on $b$ to define the polynomials in (3.5).

The first case is $a = 0$ and $b = d - 1 = m$. Then we need to determine all the $\langle \partial_u \partial_x^m, I \rangle$ with $o(u) = m$. For $u = s_j$ where $j$ is our fixed index, applying the multinomial theorem and the vanishing results, we have

$$I_j(\epsilon + s_j) = \frac{1}{d!} \langle \partial_{s_j} \partial^{d}, I_j \rangle$$

by (2.3)

$$\tag{3.7} = \frac{1}{m!} \langle \partial_{s_j} \partial_x^m, I_j \rangle.$$  

Therefore the defining condition (1.12) in this case, $I_j(\epsilon + s_j) = 1$, implies

$$\tag{3.8} \langle \partial_{s_j} \partial_x^m, I_j \rangle = m!.$$

For $u = s_i \in \mathfrak{s}$ with $o(s_i) = m$ but $i \neq j$ (hence the multiplicity of $m$ as an exponent is at least 2 and this happens, among all the simple Lie algebras, only for $D_{2n}$ and $m = 2n - 1, n \geq 2$), similarly to (3.7), the defining condition (1.12) in this case, $I_j(\epsilon + s_i) = 0$, implies

$$\tag{3.9} \langle \partial_{s_i} \partial_x^m, I_j \rangle = 0, \quad o(s_i) = o(s_j), \; i \neq j.$$

For $u = [\epsilon, v] \in [\epsilon, \mathfrak{g}]$, the vanishing property (2.10), together with (2.6) and (2.8), forces

$$\langle \partial_u \partial_x^m, I \rangle = \langle \partial_{[\epsilon, v]} \partial_x^m, I \rangle = -\frac{1}{m+1}(v \cdot \partial_x^{m+1}, I) = 0.$$

Now we compute (3.5) for $a = 0$ and all $b$. (Such expressions for (3.5) are all numbers.) If in $U$, at least one $u(i) \in [\epsilon, \mathfrak{g}]$, say $u(1) = [\epsilon, v_1]$, then by Lemma 2.15 we have, with $\tilde{U} = (u(2), \cdots, u(c))$,

$$\langle \partial_U \partial_x^m, I \rangle = \langle \partial_{u(1)} \partial_{\tilde{U}} \partial_x^m, I \rangle = \langle \partial_{[\epsilon, v_1]} \partial_{\tilde{U}} \partial_x^m, I \rangle = \frac{1}{b+1} \langle \partial_x^{b+1}(v_1 \cdot \partial_\tilde{U}), I \rangle$$

$$\tag{3.10} = \frac{1}{b+1} \left( \sum_{n=2}^c \langle \partial_x^{b+1} \partial_{u(2)} \cdots \partial_{v_1, u(n)} \cdots \partial_{u(c)} I \rangle \right),$$

where all the terms on the right have $(b + 1)$ $\epsilon$'s, and hence are known from the induction hypothesis. We need to show compatibility when there are two $u(i) \in [\epsilon, \mathfrak{g}]$, and this is done in Proposition 3.16.

On the other hand, (1.12) implies

$$\langle \partial_U \partial_x, I \rangle = 0, \quad \text{if } c \geq 2 \text{ and all the } u(i) = s_k \in \mathfrak{s}.$$

Now assume that the (3.5) have been computed when the degree in $p$ is $\leq a - 1$ with $a \geq 1$, and we compute it for degree $a$.

We in effect use the fact that every $p \in \mathfrak{h}$ is in $[\epsilon, \mathfrak{g}]$ in (1.7). Actually for $p = \sum_{i=1}^l p_i \epsilon_{a_i}$, define

$$\tag{3.12} x_p = \sum_{i=1}^l p_i \epsilon_{a_i}, \text{ then } -[\epsilon, x_p] = p.$$

Here $x_p$ can be regarded a linear function in $p$ with values in $S^1_\epsilon$. 

Then Lemma 2.15 gives
\[
\langle \partial U \partial_p^a, I \rangle = - \langle \partial U \partial_p^{a-1} \partial_{[\epsilon, x_p]}, I \rangle \\
= \frac{1}{b+1} \sum_{n=1}^{c} (\partial_{e_p}^{b+1} \partial_{u(1)} \cdots \partial_{u(n), x_p} \cdots \partial_{u(c)} \partial_p^{a-1}, I) \\
+ \frac{a-1}{b+1} (\partial_{\epsilon_p}^{b+1} \partial_{[\epsilon, x_p]} \partial_p^{a-2}, I) \\
= \frac{1}{b+1} \sum_{i=1}^{l} \sum_{n=1}^{c} p_i (\partial_{e_p}^{b+1} \partial_{u(1)} \cdots \partial_{u(n), e_{\alpha_i}} \cdots \partial_{u(c)} \partial_p^{a-1}, I) \\
+ \frac{a-1}{b+1} \sum_{i=1}^{l} p_i \alpha_i(p) (\partial_{\epsilon_p}^{b+1} \partial_{e_{\alpha_i}} \partial_p^{a-2}, I),
\]
by (3.12) and hence
\[
[p, x_p] = \sum_{i=1}^{l} p_i \alpha_i(p) e_{\alpha_i},
\]
which has degree 2 in \( p \). The factors in the first sum have degrees \( a-1 \) in \( p \), and the factors in the second sum have degrees \( a-2 \) in \( p \). Using the induction hypothesis and (3.3), all such factors can be expressed in the known cases of (3.5).

We can continue all the way until we get \( a = d \), where we have
\[
\langle \partial_d, I \rangle = (d-1) \sum_{i=1}^{l} p_i \alpha_i(p) (\partial_{e_{\alpha_i}} \partial_p^{d-2}, I).
\]
Then by (2.3), our function \( I(p) \) on \( \mathfrak{h} \) is
\[
I(p) = \frac{1}{d!} \langle \partial_d, I \rangle.
\]

Propositions 3.18 and 3.20 below prove that the \( \{I_j\}_{j=1} \) constructed this way are algebraically independent and invariant under the Weyl group when restricted to the Cartan subalgebra \( \mathfrak{h} \). \( \square \)

**Proposition 3.16.** There is compatibility when there are two choices for \( u \in [\epsilon, g] \) in (3.10).

**Proof.** Assume, for example, \( u(1) = [\epsilon, v_1], u(2) = [\epsilon, v_2] \). Then, with \( U' = (u(3), \cdots, u(c)) \),
\[
\langle \partial U' \partial_e^b, I \rangle = \langle \partial_{u(1)} \partial_{u(2)} \partial_{U'} \partial_e^b, I \rangle
\]
can be computed in two ways using \( v_1 \) or \( v_2 \) in (3.10). The first answer \( A_1 \) using \( v_1 \) is, by (2.7) and (2.6),
\[
A_1 = \frac{1}{b+1} \left( \partial_{\epsilon_p}^{b+1} (v_1 \cdot (\partial_{u(2)} \partial_{U'})), I \right) \\
= \frac{1}{b+1} \left( (\partial_{\epsilon_p}^{b+1} \partial_{[\epsilon, u(2)]} \partial_{U'}, I) + \langle \partial_{\epsilon_p}^{b+1} \partial_{u(2)} (v_1 \cdot \partial_{U'}), I \rangle \right)
\]
In this orthogonal basis, the reflection
\[\alpha \mapsto -\alpha,\]
where the first term in the second equality uses the Jacobi identity
\[[\alpha_1,\alpha_2] = [\alpha_1,\alpha_2] + [[\alpha_1,\alpha_2],\alpha_2] = [\alpha_1,\alpha_2],\]
the third equality uses Lemma 2.15 again, and the last identity uses the Lie algebra homomorphism property (2.5) (with its root in the Jacobi identity).

\[\text{Remark 3.17.} \quad \text{Furthermore when } a \geq 1, \text{ if in } U \text{ there exists a } u \in [\epsilon, g], \text{ there is an alternative approach similar to (3.10), which is compatible with (3.13), by the same reason as above.}\]

\[\text{Proposition 3.18.} \quad \text{The } I(p) \text{ on } \mathfrak{h} \text{ defined in (3.15) is invariant under the Weyl group } W.\]

\[\text{Proof.} \quad \text{Since } W \text{ is generated on } \mathfrak{h} \text{ by simple reflections } r_i \text{ through the hyperplanes defined by } \alpha_i = 0 \text{ for } 1 \leq i \leq l, \text{ we only need to prove the invariance of the function } I(p) \text{ under } r_i \text{ for any } i. \text{ Fix an } i \text{ and we omit it from the notation.}\]

\[\text{We use an orthogonal basis of } \mathfrak{h} \text{ with the first vector being } H_\alpha = H_{\alpha_i}. \text{ Then we write}\]
\[p = xH_\alpha + Y, \quad Y \perp H_\alpha \iff \alpha(Y) = 0.\]

\[\text{In this orthogonal basis, the reflection } r_i \text{ is just the transformation } x \mapsto -x, \text{ and we only need to prove that } \langle \partial^d_p, I \rangle \text{ in (3.11) is a function of } x^2. \text{ For that purpose we run decreasing induction on } k \text{ and } l \text{ to prove that in general the}\]
\[D(k, l) := \langle \partial^k_\alpha \partial^{k}_{e-\alpha} \partial^l_\lambda \partial^{d-2k-l}_p, I \rangle\]
are functions of } x^2, \text{ with } \langle \partial^d_p, I \rangle = D(0, 0).\]

\[\text{When } 2k+1 > d, D(k, l) = 0 \text{ by (2.22). When } 2k+l = d, D(k, l) = \langle \partial^k_\alpha \partial^{k}_{e-\alpha} \partial^l_\lambda, I \rangle \text{ are constants with respect to } x, \text{ since } Y \text{ doesn't involve } x.\]
Now by Lemma 2.13 and using (3.19), we have, for $d - 2k - l \geq 1$,
\[
D(k, l) = \langle \partial_e^{k} \partial_{\alpha}^{k} \partial_{\alpha}^{d-2k-l-1} (x \partial_{\alpha} + \partial_Y), I \rangle \\
= x \langle \partial_{\alpha}^{k} \partial_e^{k} \partial_e^{d-2k-l-1}, I \rangle + \langle \partial_e^{k} \partial_{\alpha}^{k} \partial_{\alpha}^{d-2k-l-1}, I \rangle \\
= x (\partial_{\alpha}^{k} \partial_e^{k} \partial_{\alpha}^{d-2k-l-1}, I) + D(k, l + 1) \\
= -\frac{1}{k + 1} \langle \partial_{\alpha}^{k+1} (e_{\alpha} \cdot (\partial_e^{k} \partial_Y \partial_{\alpha}^{d-2k-l-1})), I \rangle + D(k, l + 1) \\
= \frac{d - 2k - l - 1}{k + 1} x^2 D(k + 1, l) + D(k, l + 1),
\]

since
\[
- e_{\alpha} \cdot (\partial_e^{k} \partial_Y \partial_{\alpha}^{d-2k-l-1}) \\
= l \partial_e^{k} \partial_{\alpha}^{d-2k-l-1} + (d - 2k - l - 1) \partial_e^{k} \partial_{\alpha}^{d-2k-l-2} \\
= (d - 2k - l - 1) \alpha(p) \partial_e^{k+1} \partial_Y \partial_{\alpha}^{d-2k-l-2} 
\]
due to that $[Y, e_{\alpha}] = \alpha(Y) e_{\alpha} = 0$, $[p, e_{\alpha}] = \alpha(p) e_{\alpha}$, and $\alpha(p) = 2x$ by (3.19).

Therefore the appearance of $x$ in $D(k, l)$ is always through an $x^2$ entry. \hfill \Box

**Proposition 3.20.** The $\{I_j\}_{j=1}^l$ are algebraically independent.

**Proof.** This is clear from our defining condition (1.12), since the $I_j$ restrict to the coordinates $\xi_j$ on the slice $\epsilon + s$. \hfill \Box

**Remark 3.21.** In a sense, the above algorithm is the reversion of the procedures in [LN11] §6. Here we start with a high root vector and push the function down to $h$. In [LN11] we derived information about higher and higher root vectors starting from some knowledge on $h$. The direction here is more delicate.

**Proof of Theorem 1.17 (ii).** The further lifting of the invariant function $I = I_j$ to the whole Lie algebra $\mathfrak{g}$ involves considering all such terms
\[
(3.22) \quad \langle \partial_{W} \partial_{P} \partial_{U}, I \rangle
\]
where $\partial_{W} = \partial_{w(1)} \cdots \partial_{w(3)}$ with each $w(i)$ from $W$ in (3.4), and $\partial_{U}$ is the same as in (3.5). The absolute value of the total weight of $\partial_{W}$ is
\[
(3.23) \quad -o(W) = -\sum_{i=1}^{\beta} o(w_i).
\]

Similarly to (3.6), (3.22) is nonzero only if
\[
(3.24) \quad \beta + a + b + c = d, \quad -o(W) + b = \sum_{i=1}^{c} o(u(i)).
\]

Note that we do not need any vectors with heights $0$ or $-1$ in (3.22), since all such vectors are accounted for by the $\partial_{P}$ and $\partial_{U}$ terms using the following Lemma.
Lemma 3.25. If \( a > 0 \) and one copy of the \( p \in \mathfrak{h} \) is replaced by \( H_{\alpha_i} \), then

\[(3.26) \quad \langle \partial_W \partial_d \partial_p \partial_{H_{\alpha_i}} \partial_U, I \rangle = \frac{1}{a} \partial_{H_{\alpha_i}} \langle \partial_W \partial_d \partial_p \partial_U, I \rangle.\]

Combining the \( U \) and \( W \) in \((3.22)\) together and changing them back to the standard basis using root vectors, let \( V = \{e_{\alpha_1}, \ldots, e_{\alpha_{n}}\} \) be a sequence of (possibly repeating) root vectors from \( \bigoplus_{k > 0} \mathfrak{g}_k \oplus \bigoplus_{k \leq -2} \mathfrak{g}_k \). Let \( \alpha_V = \sum_{j=1}^{\gamma} \alpha_j = \sum_{i=1}^{l} n_i \alpha_i \) be the sum of the roots and express it in terms of the simple roots. Assume that \( n_i \geq 0 \) for \( 1 \leq i \leq l \). Let \( \partial_V = \partial_{e_{\alpha_1}} \cdots \partial_{e_{\alpha_{n}}} \), and let \( o(V) = \sum_{i=1}^{l} n_i \) be the grading of \( \partial_V \) in \( S_\ast(\mathfrak{g}) \) as defined in \((2.12)\). Let \( I \in I(\mathfrak{g}) \) be an invariant function with degree \( d \geq \gamma + o(V) \). We have

\[(3.27) \quad \langle \partial_d^{\gamma-o(V)} \partial_V \partial_p^{d-\gamma-o(V)}, I \rangle = \frac{o(V)!}{n_1! \cdots n_l!} \left( \prod_{i=1}^{l} \partial_{\alpha_i} \right) \langle \partial_V \partial_p^{d-\gamma-o(V)}, I \rangle,\]

where \( p \in \mathfrak{h} \).

If \( o(V) > 0 \) and one copy of \( e \) is replaced by \( e_{-\alpha_i} \), then

\[(3.28) \quad \langle \partial_d^{\gamma-o(V)} \partial_{e_{-\alpha_i}} \partial_V \partial_p^{d-\gamma-o(V)}, I \rangle = \frac{n_i}{o(V)} \langle \partial_d^{\gamma-o(V)} \partial_V \partial_p^{d-\gamma-o(V)}, I \rangle.\]

Proof of Lemma 3.25. Let \( p = \sum_{i=1}^{l} p_i H_{\alpha_i} \in \mathfrak{h} \). The \( \partial_{H_{\alpha_i}} \) on the right hand side of \((3.26)\) stands for \( \frac{\partial}{\partial p_i} \). All these formulas are easy consequences of the multinomial theorem for the powers of \( \partial_d \) and \( \partial_e \), and in \((3.27)\) and \((3.28)\) we also need to use Lemma 2.14. We omit the details. \( \square \)

Now returning to the main proof. We run induction on the lexicographical order of the pair \((\beta, -o(W))\), with \(-o(W)\) defined in \((3.23)\), to determine such terms in \((3.22)\). Since \( w(1) \in [\epsilon, \mathfrak{g}] \) from \((3.1)\), assume

\[(3.29) \quad w(1) = [\epsilon, v_1].\]

Then \( v_1 \in \bigoplus_{k \leq -1} \mathfrak{g}_k \). With \( \tilde{W} = (w(2), \ldots, w(\beta)) \), Lemma 2.15 gives

\[
\langle \partial_W \partial_d^{\gamma} \partial_p \partial_U, I \rangle = \langle \partial_{[\epsilon, v_1]} \partial_{\tilde{W}} \partial_d^{\gamma} \partial_p \partial_U, I \rangle
\]

\[
= \frac{1}{b+1} \sum_{m=2}^{\beta} \langle \partial_{w(2)} \cdots \partial_{[v_1, w(m)]} \cdots \partial_{w(\beta) \partial_d^{\gamma+1} \partial_p \partial_U, I} \rangle
\]

\[
+ \frac{a}{b+1} \langle \partial_{\tilde{W}} \partial_d^{\gamma+1} \partial_p^{-1} \partial_{[v_1, p]} \partial_U, I \rangle
\]

\[
+ \frac{1}{b+1} \sum_{n=1}^{c} \langle \partial_{\tilde{W}} \partial_d^{\gamma+1} \partial_p^{n} \partial_{u(1)} \cdots \partial_{[v_1, u(n)]} \cdots \partial_{u(c)} \partial_U, I \rangle.
\]

Here all the summands in the first sum can be expressed by \((3.22)\) with \( \beta - 1 \) elements from \( W \) in \((3.3)\). The summands in the second and the third sums can either be expressed by \((3.22)\) with \( \beta - 1 \) elements from \( W \) if the heights of \([v_1, p] \) or \([v_1, u(n)] \) are \( \geq -1 \) by Lemma 3.25 or with \( \beta \) elements from \( W \) otherwise. But the new \(-o([v_1, p]) \) or \(-o([v_1, u(n)]) \) is strictly less than the old \(-o(w(1)) \), since the height of \( v_1 \) is one bigger than that of \( w(1) \) in view of \((3.29)\), and \( p \) and \( u(n) \) have nonnegative heights. Therefore the new total \(-o(W) \) is strictly less than the old one.
Therefore through this hierarchy of induction hypothesis, all the terms on the right are known.

We note that the outcome of (3.30) does not depend on the choice of \( v_1 \) in (3.29), which may not be unique. Say \( v_1' = v_1 + v_0 \) with \([\epsilon, v_0] = 0\). Then the outcome of (3.30) is linear in \( v_1 \) and \( v_0 \), and the terms for \( v_0 \) combine to give

\[
\langle \partial_{[\epsilon, v_0]} \partial_{\tilde{w}} \partial_{\tilde{b}} \partial_{U} \partial_{I}, I \rangle = 0
\]

by tracing the identity backward. \( \square \)

After all these coefficients in (3.5) and (3.22), as functions on \( p \in \mathfrak{h} \), are calculated, we can assemble our function as follows. Let

\[
x = \sum_{w_i \in W} z_i w_i + \epsilon + p + \sum_{u_j \in U} y_j u_j
\]

be an element in \( \mathfrak{g} \) with the \( w_i \) from \( W \) in (3.4), the \( u_j \) from \( U \) in (3.3), \( p \in \mathfrak{h} \), and the \( z_i \) and \( y_j \) as coefficients. Then we get \( I(x) \) by (2.3), the multinomial theorem, and the coefficients (3.5) and (3.22).

If we change the basis back to the usual root vectors, then we get \( I(x) \) for \( x = p + \epsilon + \sum_{\alpha(\beta) \neq -1} x_\alpha e_\alpha \). Using (3.27), we can further spell out the dependence on the \( e_{-\alpha_\alpha} \). At the end, we obtain the function \( I(x) \) expressed in the coordinates of a general element in \( \mathfrak{g} \):

\[
x = \sum_{i=1}^{l} p_i H_{\alpha_i} + \sum_{\alpha \in \Delta} x_\alpha e_\alpha.
\]

See the end of Section 4 for an example.

4. Implementation of the algorithm on Maple

It turns out that our algorithm is very ready for implementation on Maple, especially using the LieAlgebras package under Maple written by Prof. Ian Anderson. One particularly useful feature is that we can do the change of basis in Lemma 3.2 easily. In this section we outline the steps that we run on Maple. This author has written a Maple program containing all the implementations. He is currently working with his collaborator to apply the program to the exceptional Lie algebras of the \( E \) type.

First we setup the Lie algebra \( \mathfrak{g} \) in Maple, preferably under a Chevalley basis. This is achieved by either giving a list of root vectors as matrices if one knows a representation, or by inputting the non-zero structure equations. For the exceptional Lie algebras of the \( E \) type, the structure constants for a Chevalley basis are available, for example following [Vav04]. After the DGsetup command in Maple, we have the basis vectors labeled by \( e_i \) for \( 1 \leq i \leq \text{dim} \mathfrak{g} \) and the Lie bracket operation available. The corresponding roots for the \( e_i \) are readily calculated in terms of the simple roots, and so are the heights (1.4). This author has made the choice of ordering the root vectors according to the heights from 0 to the maximal height, and then from \(-1\) to the negative maximal height.

Then we make the change of basis from the \( \{e_i\} \) to the basis in Lemma 3.2. Here we need to input the homogeneous slice basis \( \{s_j\} \) (1.7). In practice, it is sufficient to choose root vectors at each height which appears as an exponent according to its multiplicity. The \( \epsilon \) in (1.0) is readily calculated, and by inductively calculating (8.3), we get a new basis. We order these basis vectors according to their homogenous heights as before, and set up the Lie algebra again in this new
basis. This is particularly easy using the LieAlgebras package, and in particular we have the transition matrix \( M \) immediately ready. It is this matrix \( M \) that allows us to go back and forth between the two bases easily. We also record where we have the relation \( u = [e, v] \) in (3.1), and construct a procedure \( \text{fv} \) to find a preimage of an element in \( U \) (3.3) or \( W \) (3.4) under \( \text{ad}_\epsilon \).

The first big step for the program is to find all the terms (3.5) and more generally (3.22) which are possibly nonzero. In this implementation, we record each term (3.22) as a polynomial using the \( W \) and \( U \) information, and we further record its \( b \) and \( a \) values, which are dictated by (3.24).

For this we use the vanishing condition (2.13), since our new basis vectors from \( U \) in (3.3) and \( W \) in (3.4) are still homogeneous with respect to the height grading (1.5). First we compute the possible height vectors for \( W \) and \( U \), and then we readily get all the monomials corresponding to \( W \) and \( U \) with these height structures.

Let \( m_l \) (1.3) be the maximal height of the roots. Using the absolute values of the heights for \( \partial_W \) for simplicity, we are looking for partitions \( [o^W_1, o^W_2, \ldots, o^W_\beta] \) of the total height \( -o(W) \) in (3.23) into \( \beta \) parts. Here we require that \( 2 \leq o^W_i \leq m_l \) for \( 1 \leq i \leq \beta \). For the heights of \( \partial_U \), we are looking for partitions \( [o^U_1, \ldots, o^U_c] \) of the total height \( -o(W) + b \) determined by (3.24) into \( c \) parts with \( 1 \leq o^U_i \leq m_l \).

All such terms can be generated using modified partition functions by running through some loops. See Appendix A. We store the results in several lists. The list \( \text{ttms} \) stands for top-terms and corresponds to \( \beta = 0 \) and \( a = 0 \) in (3.22). The list \( \text{ptms} \) stands for \( p \)-terms with \( p \in \mathfrak{h} \) and corresponds to \( \beta = 0 \) and \( a > 0 \) in (3.22). The list \( \text{ntms} \) stands for negative-terms and corresponds to \( \beta > 0 \) in (3.22). In Appendix A, we have paid attention so that the orders of the lists follow our inductive procedures in our proofs in Section 3.

The partition function grows very fast, and this corresponds to the fact that the number \( \frac{(d+n-1)!}{d!(n-1)!} \) of terms in a degree \( d \) polynomial in \( n \) variables grows very fast with \( d \) and \( n \). Therefore it takes a huge computational time to complete this generating procedure in Appendix A when the degree \( d \) and the dimension of the Lie algebra \( n \) get big. If we are only interested in getting a function in \( \mathcal{P}(\epsilon + b)^N \) as in part (i) of Theorem 1.11, then we can let \( \beta = 0 \). The corresponding procedure is much shorter, and we still get interesting functions in \( \mathcal{P}(\epsilon + b)^N \).

The terms in (3.8), (3.9) and (3.11) are characterized by the fact all the factors in \( \partial_U \) do not come from \([e, g]\). We easily find them in our list \( \text{ttms} \) using the procedure \( \text{fv} \). Although we have specified their values in (3.8), (3.9) and (3.11), we can actually define the corresponding values as arbitrary constants, which corresponds to getting an algebraic combination of the primitive invariant functions. We store the values in a list \( \text{valuedata} \).

Then we run through our inductive algorithms for the lists \( \text{ttms}, \text{ptms} \) and \( \text{ntms} \) using the formulas (3.10), (3.13) and (3.30). Note that for the (3.30) terms, we need to take into considerations Lemma (3.25). Each time we compute a term using the previous \( \text{valuedata} \) information, and we add the result to the list \( \text{valuedata} \). These formulas are quite ready to be coded up, as this author has done. There is an extra term which consists of purely Cartan terms, and it is calculated by (3.14).

After we get all the terms stored in \( \text{valuedata} \), we can put the function together, change back the basis, and spell out the dependence on the \( e_{-\alpha_i} \) as explained at
Now we illustrate our program by the degree 6 invariant function on \( \mathfrak{g}_2 \).

We use the basis of \( \mathfrak{g}_2 \) as made explicit in the Appendix of [BFO+90]. We setup our \( \mathfrak{g}_2 \) with basis

\[
\begin{align*}
e_1 &= H_{\alpha_1} & e_2 &= H_{\alpha_2} \\
e_3 &= e_{\alpha_1} & e_4 &= e_{\alpha_2} & e_5 &= e_{\alpha_1 + \alpha_2} \\
e_6 &= e_{2\alpha_1 + \alpha_2} & e_7 &= e_{3\alpha_1 + \alpha_2} & e_8 &= e_{3\alpha_1 + 2\alpha_2} \\
e_9 &= e_{-\alpha_1} & e_{10} &= e_{-\alpha_2} & e_{11} &= e_{-\alpha_1 - \alpha_2} \\
e_{12} &= e_{-2\alpha_1 - \alpha_2} & e_{13} &= e_{-3\alpha_1 - \alpha_2} & e_{14} &= e_{-3\alpha_1 - 2\alpha_2}
\end{align*}
\]

We choose our slice elements in (1.7) to be \( s_1 = e_4, s_2 = e_8 \). Let \( \epsilon = e_9 + e_{10} \), and we do the change of basis in Lemma 3.2. Order the new basis according to our above convention, and denote them by \( \{ f_i \}_{i=1}^{14} \).

The degree \( d \) is set to be 6. A nonzero term from (3.22), for example,

\[
\langle \partial_{f_1}^2, \partial_{f_2}, \partial_{f_3}^2, I \rangle = \text{recorded by } y_{11}^2 y_8 \text{ with } b = 1 \text{ and } a = 2.
\]

We generate the possible nonzero terms by the procedure in Appendix A. There are 8 terms in \texttt{ttms}, 10 in \texttt{ptms} with one extra purely on the Cartan subalgebra (3.14), and 535 in \texttt{ntms}.

Among the terms in \texttt{ttms}, the inputs are calculated to be \( y_8 \) and \( y_3^2 \) corresponding to \( \langle \partial_{f_1}^2, \partial_{f_3}^2, I \rangle \) and \( \langle \partial_{f_2}, \partial_{f_3}^2, I \rangle \). We require that their values be 5! and 0 by (3.8) and (3.11), that is, we start our \texttt{valuedata} = \{[y_8, 240], [y_3^2, 0]\}. Running through our program we get all the values for the lists \texttt{ttms}, \texttt{ptms} and \texttt{ntms}.

For example, our algorithm and program compute that

\[
\langle \partial_{f_1}^2, \partial_{f_3}^2, \partial_{f_3}^2, I \rangle = 3136 (3p_2 - p_2)(3p_1 - 2p_2).
\]

Then by (2.3) and the multinomial theorem, \( I(p + \epsilon + \sum_{i=3}^8 y_i f_i + \sum_{j=11}^{14} y_j f_j) \) has one term

\[
\frac{1}{2! \cdot 2!} \langle \partial_{f_1}^2, \partial_{f_3}^2, \partial_{f_3}^2, I \rangle y_{11}^2 y_8 = 784 y_{11}^2 y_8 (3p_2 - p_2)(3p_1 - 2p_2).
\]

Now change the basis back to the root vectors \( e_i \), \( I(p + \epsilon + \sum_{i=3}^8 x_i e_i + \sum_{j=11}^{14} x_j e_j) \) has one term

\[
x_{11}^2 x_8 (3p_2 - p_2)(3p_1 - 2p_2),
\]

since the relevant change of basis is \( f_8 = e_8 \) and \( f_{11} = 28 e_{11} \). Finally spelling out the dependence of the function on the \( e_{-\alpha_i} \) by (3.22), we see that \( I(\sum_{i=1}^2 p_i H_{\alpha_i} + \sum_{i=3}^8 x_i e_i + \sum_{j=11}^{14} x_j e_j) \) has one term

\[
x_{11}^2 x_9 x_8 (3p_2 - p_2)(3p_1 - 2p_2),
\]

since the total root of \( \partial_{e_{11}}^2 \partial_{e_8} \) is \( 2(-\alpha_1 - \alpha_2) + (3\alpha_1 + 2\alpha_2) = \alpha_1 \), and \( e_{-\alpha_1} = e_9 \) in our ordering.

A computer program can do all the above to all the terms in \texttt{valuedata}, and eventually we obtain our function \( I(x) \) for a general \( x \in \mathfrak{g}_2 \). It turns out to be one quarter of the sum of principal minors of dimension 6 of the corresponding matrix representation of \( \mathfrak{g}_2 \). The whole procedure takes about 10 seconds on a usual laptop.
With his collaborator, the author will apply this program to the $E$'s, at least aiming to get the invariant functions on $\epsilon + b$. He believes that this program has computational advantage, since the corresponding functions are doable on a usual laptop. For example, he can compute the degree 12 invariant function on $\epsilon + b$ of $E_6$, while the usual method of taking the sum of principal minors cannot be done on a usual laptop even in the degree 5 case when using the adjoint representation.

Appendix A. Generate and order terms

We explain more the Maple codes to generate all the possibly nonzero terms in (3.22). Here $d$ is the degree of the invariant function, and $mx = m_I$ (1.3) is the maximal height of roots. The author has actually worked out in more detail the ranges for all the indices. Here floor and ceil are the standard integer functions. $\omega = -\omega(W)$ is the absolute value of the total height of $W$ (3.22). The functions negpart($x,y,z$) and pospart($x,y,z$) are the author's modified partition functions, implemented elsewhere, to partition $x$ in to $z$ parts with each part ranging from 2 for negpart (or 1 for pospart) to $y$. nwts and pwts are the absolute values of the negative and positive height vectors for $\partial W$ and $\partial U$ in (3.22). Also allmul is the author's procedure, implemented elsewhere, to come up with the terms such as (4.1) in (3.22) from the height vectors.

```
ttms:=[ ]: ptms:=[ ]: ntms:=[ ]:
for beta from 0 to floor($d*mx/(mx+2)$) do
  for ow from 2*beta to min(beta, $d-beta$)*mx do
    nwts := negpart(ow, mx, beta):
    for a from 0 to $d-beta-1$ do
      for b from max(0, ceil($((d-beta-a)-ow)/2$)) to floor($((d-beta-a)*mx-ow)/(mx+1)$) do
        pwts := pospart(ow+b, mx, $d-beta-a-b$):
        if beta=0 then
          if a=0 then
            ttms := [seq(seq([a, b, f], f in allmul(ls)), ls in pwts), op(ttms)]:
          else
            ptms := [op(ptms), seq(seq([a, b, f], f in allmul(ls)), ls in pwts)]:
          fi:
        else
          ntms := [op(ntms), seq(seq(seq(seq([beta, nt, f*g, a, b], g in allmul(ls)), f in allmul(-nt)), ls in pwts), nt in nwts)]:
          fi:
      od:
  od:
od: od: od: od:
```

References

[BFO+90] J. Balog, L. Fehér, L. O’Raifeartaigh, P. Forgács, and A. Wipf, *Toda theory and W-algebra from a gauged WZNW point of view*, Ann. Physics 203 (1990), no. 1, 76–136.

[Che52] C. Chevalley, *The Betti numbers of the exceptional simple Lie groups*, Proceedings of the International Congress of Mathematicians, Cambridge, Mass., 1950, vol. 2, Amer. Math. Soc., Providence, R. I., 1952, pp. 21–24.

[CP07] Martin Cederwall and Jakob Palmkvist, *The octic $E_8$ invariant*, J. Math. Phys. 48 (2007), no. 7, 073505, 7pp.

[Che55] Claude Chevalley, *Invariants of finite groups generated by reflections*, Amer. J. Math. 77 (1955), 778–782.
[GS99] M. I. Gekhtman and M. Z. Shapiro, *Noncommutative and commutative integrability of generic Toda flows in simple Lie algebras*, Comm. Pure Appl. Math. 52 (1999), no. 1, 53–84.

[Kos59] Bertram Kostant, *The principal three-dimensional subgroup and the Betti numbers of a complex simple Lie group*, Amer. J. Math. 81 (1959), 973–1032.

[Kos63] ———, *Lie group representations on polynomial rings*, Amer. J. Math. 85 (1963), 327–404.

[Kos78] ———, *On Whittaker vectors and representation theory*, Invent. Math. 48 (1978), no. 2, 101–184.

[Kos79] ———, *The solution to a generalized Toda lattice and representation theory*, Adv. in Math. 34 (1979), no. 3, 195–338, DOI 10.1016/0001-8708(79)90057-4. MR550790 (82f:58045)

[LN11] Luen-Chau Li and Zhaohu Nie, *Liouville integrability of a class of integrable spin Calogero-Moser systems and exponents of simple Lie algebras*, Communications in Mathematical Physics 308 (2011), no. 2, 415–438.

[Vav04] N. A. Vavilov, *Do it yourself structure constants for Lie algebras of types $E_l$*, J. Math. Sci. (N. Y.) 120 (2004), no. 4, 1513–1548 (English translation).

E-mail address: zhaohu.nie@usu.edu

Department of Mathematics and Statistics, Utah State University, Logan, UT 84322-3900