WEIGHT MULTIPLICITY FORMULAS FOR BIVARIATE REPRESENTATIONS OF CLASSICAL LIE ALGEBRAS

EMILIO A. LAURET, FIORELA ROSSI BERTONE

ABSTRACT. A bivariate representation of a complex simple Lie algebra is an irreducible representation having highest weight a combination of the first two fundamental weights. For a complex classical Lie algebra, we establish an expression for the weight multiplicities of bivariate representations.

1. INTRODUCTION

This article concerns on giving weight multiplicity formulas, continuing the previous authors’ article [LR17]. In that article, it was determined, for a classical complex Lie algebra $\mathfrak{g}$, a closed explicit formula for the weight multiplicities of any representation of any $p$-fundamental string. Such a representation is an irreducible representation of $\mathfrak{g}$ with highest weight $k\omega_1 + \omega_p$ for some non-negative integer $k$. Here, $\omega_j$ denotes the $j$th fundamental weight associated with the root system of $\mathfrak{g}$.

The primary goal of the present article is to find an expression for the weight multiplicity of every bivariate representation of a classical complex Lie algebra $\mathfrak{g}$. A bivariate representation is an irreducible representation with highest weight $a\omega_1 + b\omega_2$ for some non-negative integers $a$ and $b$ (cf. [Ma14]). See [LR17, §1] for references of classical and recent previous results on this problem.

In Section 2 we introduce the standard notation used to describe the root system associated to a classical complex Lie algebra $\mathfrak{g}$. In particular, for $\mathfrak{g}$ of type $B_n$, $C_n$ or $D_n$ and $\mathfrak{h}$ a fixed Cartan subalgebra of $\mathfrak{g}$, $\{\varepsilon_1, \ldots, \varepsilon_n\}$ denotes the basis of $\mathfrak{h}^*$ satisfying that the set of simple roots are $\{\varepsilon_1 - \varepsilon_2, \ldots, \varepsilon_{n-1} - \varepsilon_n, \varepsilon_n\}$ for type $B_n$, $\{\varepsilon_1 - \varepsilon_2, \ldots, \varepsilon_{n-1} - \varepsilon_n, 2\varepsilon_n\}$ for type $C_n$, and $\{\varepsilon_1 - \varepsilon_2, \ldots, \varepsilon_{n-1} - \varepsilon_n, \varepsilon_n + 1 + \varepsilon_n\}$ for type $D_n$. According to this notation, bivariate representations have highest weight of the form $k\varepsilon_1 + l\varepsilon_2$ for integers $k \geq l \geq 0$.

The obtained weight multiplicity formulas for types $B_n$, $C_n$ and $D_n$ are in Theorems 3.1, 3.2 and 3.3 respectively. The expressions involve a sum over partitions of the integer numbers $\leq l$, so they may not be considered ‘closed explicit formulas’ like in [LR17]. An immediate and curious consequence of the formulas is the next result.

Theorem 1.1. Let $\mathfrak{g}$ be a classical complex Lie algebra of type $B_n$, $C_n$ or $D_n$. Let $k \geq l \geq 0$ integers and $\mu = \sum_{i=1}^{n} a_i\varepsilon_i$ with $a_i \in \mathbb{Z}$ for all $i$. The multiplicity of $\mu$ in the irreducible representation $\pi_{k\varepsilon_1+l\varepsilon_2}$ of $\mathfrak{g}$ with highest weight $k\varepsilon_1 + l\varepsilon_2$, denoted by $m_{\pi_{k\varepsilon_1+l\varepsilon_2}}(\mu)$, depends only on

\[ \|\mu\|_1 := \sum_{i=1}^{n} |a_i| \quad \text{and} \quad \ell_t(\mu) := \# \{ i : 1 \leq i \leq n, |a_i| = t \} \quad \text{for all } 0 \leq t \leq l - 1. \]

In other words, if $\mu$ and $\mu'$ satisfy $\|\mu\|_1 = \|\mu'\|_1$ and $\ell_t(\mu) = \ell_t(\mu')$ for all $0 \leq t \leq l - 1$, then $m_{\pi_{k\varepsilon_1+l\varepsilon_2}}(\mu) = m_{\pi_{k\varepsilon_1+l\varepsilon_2}}(\mu')$.
This theorem is analogous to [LR17, Cor. 1.1] (see also [LMR16, Lem. 3.3]), which states that the multiplicity of a weight \( \mu \) in representations in \( p \)-fundamental strings depends only on \( \| \mu \|_1 \) and \( \ell_0(\mu) \). Such representations have highest weights of the form \( k\omega_1 + \omega_p \) for \( k \geq 0 \) and, \( 1 \leq p \leq n-1 \) for type \( B_n \), \( 1 \leq p \leq n \) for type \( C_n \), \( 1 \leq p \leq n-2 \) for type \( D_n \).

In the best authors’ knowledge, the weight multiplicity formulas in Theorems 3.1–3.3 are not in the literature. Nevertheless, Maddox [Ma14] obtained a multiplicity formula for bivariate representations when \( g \) is of type \( C_n \). However, her expression differs significantly from ours. In particular, Theorem 1.1 does not follow immediately from her formula. See Remark 3.14 for more details.

We compare from a computational point of view, the multiplicity formulas obtained in Theorems 3.1–3.3 with Freudenthal’s famous formula (see Subsection 3.3). We used the open-source mathematical software Sage [Sa] to do the calculations. It was evidenced in the computational results shown in Table 1 that the bivariate algorithm based on Theorems 3.1–3.3 is faster than Freudenthal algorithm for most of small values of \( k \) and \( l \). Moreover, for any choice of \( k \) and \( l \), the same conclusion would hold for \( n \) big enough. It is probably more significant the fact that Theorems 3.1–3.3 return in a speedy way the multiplicity of a single weight. The situation is very different with Freudenthal’s formula since it is defined recursively, and moreover, it has to calculate the multiplicities of many intermediate weights in case the original weight is far away from the highest weight. Many more related remarks are made in Subsection 3.3.

We have already mentioned that the expressions for the weight multiplicities in Theorems 3.1–3.3 and 4.2 are not closed explicit formulas since they involve a sum over partitions. However, in some particular cases, one can write down the corresponding partitions obtaining a (long) closed expression. For instance, the statements for the multiplicity of the weight \( \mu = 0 \) for types \( B_n \), \( C_n \) and \( D_n \) are included in Subsection 3.4. Furthermore, a formula for \( \pi_{k\varepsilon_1 + 2\varepsilon_2} \) in type \( D_n \) is also part of that subsection.

Although weight multiplicity formulas are interesting in themselves, the authors were motivated by their application in spectral geometry (see [LR17, §7]). In Remark 3.17 we mention possible applications for the weight multiplicity formulas obtained in this article for the determination of the spectra of some natural differential operators on spaces covered by compact symmetric spaces with abelian fundamental groups.

The weight multiplicity formula for \( g \) of type \( A_n \) is determined in Section 4. Furthermore, the corresponding expression for the case \( l = 2 \) is stated in Corollary 4.3. This case, \( g \) of type \( A_n \), is much simpler than the previous ones. The obtained expressions are probably already present in the extensive literature on this area.

The article is organized as follows. Section 2 introduces the necessary notation to read the statements of the primary results. Section 3 which considers classical Lie algebras of type \( B_n \), \( C_n \) and \( D_n \), is divided in five subsections. The first one states the weight multiplicity formulas which are proven in the second one. The computational comparison is made in Subsection 3.3. Subsection 3.4 shows closed explicit formulas in particular cases. Section 3 ends with some remarks. The case when \( g \) is of type \( A_n \) is considered in Section 4.

2. Notation

Throughout this section, \( g \) denotes a classical complex Lie algebra of type \( B_n \), \( C_n \) and \( D_n \), namely \( \mathfrak{so}(2n + 1, \mathbb{C}) \), \( \mathfrak{sp}(n, \mathbb{C}) \), \( \mathfrak{so}(2n, \mathbb{C}) \) respectively. We assume \( n \geq 2 \) for types \( B_n \) and \( C_n \), and \( n \geq 3 \) for \( D_n \). We fix a Cartan subalgebra \( \mathfrak{h} \) of \( g \).
Let \( \{\varepsilon_1, \ldots, \varepsilon_n\} \) be the standard basis of \( \mathfrak{h}^* \). Then, the sets of positive roots \( \Sigma^+(\mathfrak{g}, \mathfrak{h}) \) and the space of integral weights \( \mathcal{P}(\mathfrak{g}) \) are respectively given by

\[
\begin{aligned}
\{\varepsilon_i \pm \varepsilon_j : i < j\} \cup \{\varepsilon_i\} & \quad \text{and} \quad \{\sum_i a_i \varepsilon_i : a_i \in \mathbb{Z} \forall i, \text{ or } a_i = -\frac{1}{2} \in \mathbb{Z} \forall i\} \quad \text{for } \mathfrak{g} \text{ of type } B_n, \\
\{\varepsilon_i \pm \varepsilon_j : i < j\} \cup \{2\varepsilon_i\} & \quad \text{and} \quad \{\sum_i a_i \varepsilon_i : a_i \in \mathbb{Z} \forall i\} \quad \text{for } \mathfrak{g} \text{ of type } C_n, \\
\{\varepsilon_i \pm \varepsilon_j : i < j\} & \quad \text{and} \quad \{\sum_i a_i \varepsilon_i : a_i \in \mathbb{Z} \forall i, \text{ or } a_i = -\frac{1}{2} \in \mathbb{Z} \forall i\} \quad \text{for } \mathfrak{g} \text{ of type } D_n.
\end{aligned}
\]

Furthermore, \( \sum_i a_i \varepsilon_i \in \mathcal{P}(\mathfrak{g}) \) is dominant if and only if \( a_1 \geq a_2 \geq \cdots \geq a_n \geq 0 \) for types \( B_n \) and \( C_n \) and \( a_1 \geq \cdots \geq a_{n-1} \geq |a_n| \) for type \( D_n \).

By the Highest Weight Theorem, irreducible representations of \( \mathfrak{g} \) are in correspondence with integral dominant weights. We denote by \( \mathcal{P}^{++}(\mathfrak{g}) \) the set of integral dominant weights and by \( \pi_\lambda \) the irreducible representation of \( \mathfrak{g} \) with highest weight \( \lambda \in \mathcal{P}^{++}(\mathfrak{g}) \).

The first two fundamental weights are \( \omega_1 = \varepsilon_1 \) and \( \omega_2 = \varepsilon_1 + \varepsilon_2 \). Hence, any non-negative integer combination of them is of the form \( k\varepsilon_1 + l\varepsilon_2 \) for some integers \( k \geq l \geq 0 \).

The following notation is essential to read the weight multiplicity formulas in the next section. We will write \( \mathbb{Z}^n \) for the set of elements \( \sum_i a_i \varepsilon_i \in \mathcal{P}(\mathfrak{g}) \) such that \( a_i \in \mathbb{Z} \) for all \( i \). For a weight \( \mu = \sum_{i=1}^n a_i \varepsilon_i \in \mathbb{Z}^n \) and \( t \) a non-negative integer, define

\[
(2.1) \quad \|\mu\|_1 = \sum_{i=1}^n |a_i|, \quad \ell_t(\mu) = \#\{i : 1 \leq i \leq n, |a_i| = t\}.
\]

Given \( N \geq 0 \), let \( \mathcal{Q}_n(N) \) be the set of all partitions of \( N \) with length \( \leq n \), that is

\[
(2.2) \quad \mathcal{Q}_n(N) = \{\mathbf{q} = (q_1, q_2, \ldots, q_n) \in \mathbb{Z}^n : q_1 \geq q_2 \geq \cdots \geq q_n \geq 0, \sum_{i=1}^n q_i = N\}.
\]

Furthermore, for \( \mathbf{q} \in \mathcal{Q}_n(N) \) and \( 1 \leq j \leq N \), we set

\[
(2.3) \quad s_j^\mathbf{q} := \#\{i : 1 \leq i \leq n, q_i = j\}, \\
(2.4) \quad \mathcal{B}_\beta := \{(\beta_1^1, \beta_2^2, \beta_3^3, \ldots, \beta_N^N) : \beta_i^j \geq 0, \sum_{i=1}^j \beta_i^j \leq s_j^\mathbf{q}\}, \\
(2.5) \quad \mathcal{A}_\beta := \{(\alpha_1^1, \alpha_2^2, \alpha_3^3, \ldots, \alpha_N^N) : 0 \leq \alpha_i^j \leq \beta_i^j\} \quad \text{for any } \beta \in \mathcal{B}_\beta.
\]

Throughout the article we use the convention \( \binom{b}{a} = 0 \) if \( a < 0 \) or \( b < a \).

### 3. Types \( B_n \), \( C_n \) and \( D_n \)

In this section we consider \( \mathfrak{g} \) a classical complex Lie algebra of types \( B_n, C_n \) and \( D_n \). We assume that \( n \geq 2 \) for types \( B_n \) and \( C_n \) and \( n \geq 3 \) for type \( D_n \).

#### 3.1. Main results

We now state the three theorems which establish the weight multiplicity formulas for types \( B_n \), \( C_n \) and \( D_n \) respectively. The notation required was introduced in the previous section. The formulas consider weights in \( \mathbb{Z}^n \), since the multiplicity in \( \pi_{k\varepsilon_1+l\varepsilon_2} \) of any weight in \( \mathcal{P}(\mathfrak{g}) \setminus \mathbb{Z}^n \) vanishes (see Remark 3.4).

**Theorem 3.1.** (Type \( B_n \)) Let \( \mathfrak{g} = \mathfrak{so}(2n+1, \mathbb{C}) \) for some \( n \geq 2 \). Let \( k \geq l \geq 0 \) integers and \( \mu \in \mathbb{Z}^n \). If \( r(\mu) := (k + l - \|\mu\|_1)/2 \) is negative, then \( m_{\pi_{k\varepsilon_1+l\varepsilon_2}}(\mu) = 0 \), and otherwise

\[
m_{\pi_{k\varepsilon_1+l\varepsilon_2}}(\mu) = \begin{cases} 
B_n(l, r(\mu), \ell_0(\mu), \ldots, \ell_{l-1}(\mu)) \\
- B_n(l-1, r(\mu), \ell_0(\mu), \ldots, \ell_{l-1}(\mu)) \\
- B_n(l-1, r(\mu) - 1, \ell_0(\mu), \ldots, \ell_{l-1}(\mu)) \\
+ B_n(l-2, r(\mu) - 1, \ell_0(\mu), \ldots, \ell_{l-1}(\mu)),
\end{cases}
\]

where \( \ell_0(\mu) := \ell_0, \ell_1(\mu) := \ell_1, \ldots, \ell_{l-1}(\mu) := \ell_{l-1} \).
where

\[
B_n(l, r, \ell_0, \ldots, \ell_{l-1}) = \sum_{0 \leq N \leq l} \sum_{q \in \mathcal{Q}_n(N)} \sum_{\beta \in B_\gamma} \sum_{\alpha \in A_\beta^q} \left( \frac{[(l - N)/2] + n - 1}{n - 1} \right)^n
\]
\[
\left( [r - (l + N)/2] + \sum_{j=1}^N \sum_{i=1}^j (j + 1 - i)\alpha_i \right) + n - 1
\]
\[
\prod_{j=1}^N \left( 2^{s_j - \sum_{i=1}^j \beta_i} \left( \frac{\beta_1^j}{\alpha_i} \right) \left( n - \sum_{i=0}^{j-1} \ell_i - \sum_{r=j+1}^N \sum_{s=1}^{r+j-1} \beta_s^r \right) \right)
\]
\[
\left( \ell_0 - \sum_{h=j+1}^N \left( \sum_{s=1}^{h} \beta_s^h \right) \right) \prod_{i=2}^j \left( \ell_{j-i+1} - \sum_{t=1}^{N-j} \beta_{i+t}^j \right) \left( \frac{\beta_1^j}{\alpha_i^j} \right).
\]

**Theorem 3.2.** (Type C) Let \( g = sp(n, \mathbb{C}) \) for some \( n \geq 2 \). Let \( k \geq l \geq 0 \) integers and \( \mu \in \mathbb{Z}^n \). If \( r(\mu) := (k + l - \|\mu\|_1)/2 \) is not a non-negative integer, then \( m_{\pi_{k+1}l+2}(\mu) = 0 \), and otherwise

\[
m_{\pi_{k+1}l+2}(\mu) = \begin{cases} 
C_n(l, r(\mu), \ell_0(\mu), \ldots, \ell_{l-1}(\mu)) \\
C_n(l - 1, r(\mu), \ell_0(\mu), \ldots, \ell_{l-1}(\mu)) \\
C_n(l - 1, r(\mu) - 1, \ell_0(\mu), \ldots, \ell_{l-1}(\mu)) \\
C_n(l - 2, r(\mu) - 1, \ell_0(\mu), \ldots, \ell_{l-1}(\mu)),
\end{cases}
\]

where

\[
C_n(l, r, \ell_0, \ldots, \ell_{l-1}) = \sum_{0 \leq N \leq l} \sum_{q \in \mathcal{Q}_n(N)} \sum_{\beta \in B_\gamma} \sum_{\alpha \in A_\beta^q} \left( \frac{[(l - N)/2] + n - 1}{n - 1} \right)^n
\]
\[
\left( [r - (l + N)/2] + \sum_{j=1}^N \sum_{i=1}^j (j + 1 - i)\alpha_i \right) + n - 1
\]
\[
\prod_{j=1}^N \left( 2^{s_j - \sum_{i=1}^j \beta_i} \left( \frac{\beta_1^j}{\alpha_i} \right) \left( n - \sum_{i=0}^{j-1} \ell_i - \sum_{r=j+1}^N \sum_{s=1}^{r+j-1} \beta_s^r \right) \right)
\]
\[
\left( \ell_0 - \sum_{h=j+1}^N \left( \sum_{s=1}^{h} \beta_s^h \right) \right) \prod_{i=2}^j \left( \ell_{j-i+1} - \sum_{t=1}^{N-j} \beta_{i+t}^j \right) \left( \frac{\beta_1^j}{\alpha_i^j} \right).
\]

**Theorem 3.3.** (Type D) Let \( g = so(2n, \mathbb{C}) \) for some \( n \geq 3 \). Let \( k \geq l \geq 0 \) integers and \( \mu \in \mathbb{Z}^n \). If \( r(\mu) := (k + l - \|\mu\|_1)/2 \) is not a non-negative integer, then \( m_{\pi_{k+1}l+2}(\mu) = 0 \), and otherwise

\[
m_{\pi_{k+1}l+2}(\mu) = \begin{cases} 
D_n(l, r(\mu), \ell_0(\mu), \ldots, \ell_{l-1}(\mu)) \\
D_n(l - 1, r(\mu), \ell_0(\mu), \ldots, \ell_{l-1}(\mu)) \\
D_n(l - 1, r(\mu) - 1, \ell_0(\mu), \ldots, \ell_{l-1}(\mu)) \\
D_n(l - 2, r(\mu) - 1, \ell_0(\mu), \ldots, \ell_{l-1}(\mu)),
\end{cases}
\]
where

\[ D_n(l, r, \ell_0, \ldots, \ell_{l-1}) = \sum_{\alpha \in \mathbb{N}^2 \setminus \mathbb{Z}^2} \sum_{q \in \mathcal{Q}_n(N)} \sum_{\beta \in \mathcal{B}_q} \left( \sum_{\alpha \in \mathcal{A}_q^\beta} \frac{(l - N)/2 + n - 2}{n - 2} \epsilon^{\epsilon} \right) \]

\[ \left( r - (l + N)/2 + \sum_{j=1}^{N} \sum_{i=1}^{j} (j + 1 - i) \alpha_i + n - 2 \right) \]

\[ \prod_{j=1}^{N} \left( 2 \epsilon^{\epsilon} \sum_{i=1}^{j} \beta_i \right) \left( \alpha_1^{\epsilon} \right) \left( n - \sum_{i=0}^{j-1} \ell_i - \sum_{r=j+1}^{N} \sum_{s=1}^{r-j+1} \beta_s^{\epsilon} \right) \]

\[ \left( \ell_0 - \sum_{h=j+1}^{N} \left( \frac{s_h - \sum_{s=1}^{h} \beta_s^{\epsilon}}{s_j^{\epsilon} - \sum_{t=1}^{j} \beta_t^{\epsilon}} \right) \right) \prod_{i=2}^{j} \left( \ell_{j+i-1} - \sum_{s=1}^{N-j} \beta_{s+1}^{\epsilon} \right) \left( \beta_i^{\epsilon} \right) \right). \]

**Remark 3.4.** For all types considered, it turns out that \( m_{\pi_{k\epsilon_1+l\epsilon_2}}(\mu) = 0 \) for all \( \mu \in P(\mathfrak{g}) \setminus \mathbb{Z}^n \). Indeed, the subset \( \mathbb{Z}^n \) coincides with the set of \( G \)-integral weights, where \( G \) is the only compact linear group \( G \) whose Lie algebra is a compact real form in \( \mathfrak{g} \). We have that \( G \) is isomorphic to \( \text{SO}(2n+1), \text{Sp}(n) \) and \( \text{SO}(2n) \) for types \( B_n, C_n \) and \( D_n \) respectively. Since \( k \epsilon_1 + l \epsilon_2 \in \mathbb{Z}^n \), the representation \( \pi_{k\epsilon_1+l\epsilon_2} \) descends to a representation of \( G \), and consequently their weights are in \( \mathbb{Z}^n \) (see [Kn] Lem. 5.106)).

### 3.2. Proofs

This subsection contains a unified proof of Theorems 3.1, 3.3. We first need two lemmas. The first one gives well-known closed explicit formulas for the weight multiplicities of representations having highest weights of the form \( k\epsilon_1 \) for \( k \) a non-negative integer. A proof can be found in [LR17] Lemmas 3.2, 4.3, 5.3]. The second lemma is the first step to prove the theorems.

**Lemma 3.5.** Let \( \mathfrak{g} \) be a complex Lie algebra of type \( B_n, C_n \) or \( D_n \) for some \( n \geq 2 \). Let \( k \geq 0 \) integer and \( \mu \in \mathbb{Z}^n \). Then

\[ m_{\pi_{k\epsilon_1}}(\mu) = \begin{cases} (r(\mu) + n - 1) / n & \text{for } \mathfrak{g} \text{ of type } B_n, \\ 0 & \text{for } \mathfrak{g} \text{ of type } C_n, \\ (r(\mu) + n - 2) / n & \text{for } \mathfrak{g} \text{ of type } D_n. \end{cases} \]

**Lemma 3.6.** Let \( \mathfrak{g} \) be a classical Lie algebra of type \( B_n, C_n \) or \( D_n \). For integers \( k \geq l \geq 0 \) write \( \tau_{k,l} = \pi_{k\epsilon_1} \otimes \pi_{l\epsilon_1} \). Then, in the virtual ring of representations of \( \mathfrak{g} \), we have that

\[ \pi_{k\epsilon_1+l\epsilon_2} \simeq \tau_{k,l} - \tau_{k+1,l-1} - \tau_{k-1,l+1} + \tau_{k,l-2}. \]

**Proof.** We have the fusion rule (see for instance [KT87], page 510, example (2))

\[ \tau_{k,l} = \pi_{k\epsilon_1} \otimes \pi_{l\epsilon_1} \simeq \bigoplus_{j=0}^{l} \bigoplus_{i=0}^{j} \pi_{(k+l-j-i)\epsilon_1+(j-i)\epsilon_2}. \]
As an immediate consequence we obtain
\[
\tau_{k,l} - \tau_{k+1,l-1} = \sum_{j=0}^{l} \sum_{i=0}^{j} \pi_{(k+l-j-i)\xi_1 + (j-i)\xi_2} - \sum_{j=0}^{l-1} \sum_{i=0}^{j} \pi_{(k+l-j-i)\xi_1 + (j-i)\xi_2} = \sum_{i=0}^{l} \pi_{(k-i)\xi_1 + (l-i)\xi_2};
\]
\[
\tau_{k-1,l-1} - \tau_{k,l-2} = \sum_{j=0}^{l-1} \sum_{i=0}^{j} \pi_{(k+l-j-i-2)\xi_1 + (j-i)\xi_2} - \sum_{j=0}^{l-2} \sum_{i=0}^{j} \pi_{(k+l-j-i-2)\xi_1 + (j-i)\xi_2} = \sum_{i=0}^{l} \pi_{(k-i)\xi_1 + (l-i)\xi_2}.
\]
Subtracting the previous identities we obtain the desired formula.

Proofs of Theorems 3.1–3.3. Without loss of generality we can assume that \(\mu \in \mathbb{Z}^n\) is dominant since the Weyl group preserves weight multiplicities. Recall that \(\tau_{k,l} = \pi_{k\xi} \otimes \pi_{l\xi}\) for \(k \geq l \geq 0\) integers. Since
\[
m_{\pi_{k\xi_1 + l\xi_2}}(\mu) = m_{\tau_{k,l}}(\mu) - m_{\tau_{k+1,l-1}}(\mu) - m_{\tau_{k-1,l-1}}(\mu) + m_{\tau_{k,l-2}}(\mu)
\]
by Lemma 3.6 we are left with the task of showing that
\[
m_{\tau_{k,l}}(\mu) = \begin{cases} B_n(l, r(\mu), \ell_0(\mu), \ldots, \ell_{l-1}(\mu)) & \text{for } g \text{ of type } B_n, \\ C_n(l, r(\mu), \ell_0(\mu), \ldots, \ell_{l-1}(\mu)) & \text{for } g \text{ of type } C_n, \\ D_n(l, r(\mu), \ell_0(\mu), \ldots, \ell_{l-1}(\mu)) & \text{for } g \text{ of type } D_n. \end{cases}
\]
It is well known that (see [Kn], Exercise V.14)
\[
m_{\tau_{k,l}}(\mu) = \sum_{\eta} m_{\pi_{k\xi_1}}(\mu - \eta) m_{\pi_{l\xi_1}}(\eta),
\]
where the sum is restricted to \(\mathcal{P}(\pi_{l\xi_1})\), the set of weights of \(\pi_{l\xi_1}\). By Lemma 3.3 the weights of \(\pi_{l\xi_1}\) are those \(\eta\) such that \(l - \|\eta\|_1 \in \mathbb{N}_0\) for type \(B_n\) and \(l - \|\eta\|_1 \in 2\mathbb{N}_0\) for types \(C_n\) and \(D_n\). In order to calculate \(\|\mu - \eta\|_1\), to determine \(m_{\pi_{k\xi}}(\mu - \eta)\), we make a convenient partition of \(\mathcal{P}(\pi_{l\xi_1}).\)

We write \(\ell_t = \ell_t(\mu)\) for all \(0 \leq t \leq l - 1\). For \(0 \leq N \leq l\) integer, \(q = (q_1, \ldots, q_n) \in \mathcal{Q}_n(N)\), \(\beta = (\beta_1, \beta_2, \ldots, \beta_N) \in \mathcal{B}^q\) and \(\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N) \in \mathcal{A}_\beta^q\) (see Section 2 for notation), we set
\[
P_{\beta, \alpha}^q = \left\{ \sum_{j=1}^{N} \sum_{k=0}^{s_j^q} b_j, k \varepsilon_{i_k}^j : \sum_{t=0}^{j-2} \ell_t < \cdots < \sum_{t=0}^{j-1} \beta_t^j + 1 < \cdots < \sum_{t=0}^{h} \beta_t^j, \quad \sum_{t=0}^{j-2} \ell_t < \cdots < \ell_0 + \sum_{t=0}^{h} \beta_t^j \right\}
\]
We now list some properties shared by all the elements in \(P_{\beta, \alpha}^q\). Let \(\eta = \sum_{i=1}^{n} c_i \varepsilon_i \in P_{\beta, \alpha}^q\). The multiset (i.e. a set where an element can be repeated) given by the elements \(|c_1|, |c_n|\) coincides with the multiset of elements \(q_1, \ldots, q_n\), thus \(\|\eta\|_1 = N\). For a fixed \(1 \leq j \leq N\), the number of entries equal to \(\pm j\) is \(s_j^q\) located as follows: we divide the integral interval \([1, n]\) in \((j + 1)\)-blocks as the identity \(n = (n - \sum_{t=0}^{j-1} \ell_t) + \ell_{j-1} + \ell_{j-2} + \cdots + \ell_1 + \ell_0\) suggests, that is, the first block has the first \((n - \sum_{t=0}^{j-1} \ell_t)\) integers, the second block has the next \(\ell_{j-1}\) elements, the third one has the next \(\ell_{j-2}\) elements, and so on. For each \(1 \leq t \leq j\), there are \(\beta_t^j\) entries
in the \(t\)th block equal to \(\pm j\), \(\alpha_i^j\) of them are positive. In the last block there are \(s_j^q - \sum_{t=1}^{j-1} \beta_i^t\) entries equal to \(\pm j\).

As a consequence of the previous paragraph, we have partitioned the set of weights of \(\pi_{l\epsilon_1}\) as

\[
\mathcal{P}(\pi_{l\epsilon_1}) = \bigcup_{N} \bigcup_{\alpha \in \mathcal{A}^q_{\beta}} \bigcup_{\beta \in \mathcal{B}^q} \bigcup_{\eta \in \mathcal{P}^q_{\beta,\alpha}} \mathcal{P}^q_{\beta,\alpha}
\]

where the first union is over \(N \in \mathbb{N}_0\) satisfying \(l - N \in \mathbb{N}_0\) for type \(B_n\) and \(l - N \in 2\mathbb{N}_0\) for types \(C_n\) and \(D_n\). All the unions are disjoint.

Fix an integer \(0 \leq N \leq l\), \(\eta \in \mathcal{Q}_n(N)\), \(\beta \in \mathcal{B}^q\), \(\alpha \in \mathcal{A}^q_{\beta}\) and \(\eta \in \mathcal{P}^q_{\beta,\alpha}\). One may check that

\[
\|\mu - \eta\|_1 = k + l - 2r + \sum_{j=1}^{N} \sum_{i=1}^{j} \left(j(\beta_i^j - \alpha_i^j) + (2i - j)\alpha_i^j\right) + \sum_{j=1}^{N} \left(s_j^q - \sum_{i=1}^{j} \beta_i^j\right)
\]

\[
= k + l - 2r + \sum_{j=1}^{N} \sum_{i=1}^{j} 2(i - j - 1)\alpha_i^j + j s_j^q
\]

\[
= k + l + N - 2\left(r + \sum_{j=1}^{N} \sum_{i=1}^{j} (j + 1 - i)\alpha_i^j\right)
\]

\[
= k - 2\left(r + \sum_{j=1}^{N} \sum_{i=1}^{j} (j + 1 - i)\alpha_i^j - (l + N)/2\right).
\]

Since \(m_{\pi_{l\epsilon_1}}(\eta)\) and \(m_{\pi_{k\epsilon_1}}(\mu - \eta)\) are given in Lemma 3.3 in terms of \(l - \|\eta\|_1\) and \(k - \|\mu - \eta\|_1\) respectively, it follows that \(m_{\pi_{l\epsilon_1}}(\eta)\) and \(m_{\pi_{k\epsilon_1}}(\mu - \eta)\) are constant, independent of the choice of \(\eta \in \mathcal{P}^q_{\beta,\alpha}\).

From the above fact, the partition (3.8) and the formula (3.6) we conclude that

\[
m_{\pi_{k,l}}(\mu) = \sum_{N} \sum_{\eta \in \mathcal{Q}_n(N)} \sum_{\beta \in \mathcal{B}^q} \sum_{\alpha \in \mathcal{A}^q_{\beta}} m_{\pi_{k\epsilon_1}}(\mu - \eta^q_{\beta,\alpha}) m_{\pi_{l\epsilon_1}}(\eta_{\beta,\alpha}) \#\mathcal{P}^q_{\beta,\alpha},
\]

where \(\eta^q_{\beta,\alpha}\) is any element in \(\mathcal{P}^q_{\beta,\alpha}\), and the first sum is over \(N \in \mathbb{N}_0\) satisfying \(l - N \in \mathbb{N}_0\) for type \(B_n\) and \(l - N \in 2\mathbb{N}_0\) for types \(C_n\) and \(D_n\).

By tedious but straightforward combinatorial arguments we have

\[
\#\mathcal{P}^q_{\beta,\alpha} = 2^{\sum_{j=1}^{N}(\ell_j^q - \sum_{i=1}^{j} \beta_i^j)} \prod_{j=1}^{N} \left(\sum_{t=0}^{j-1} \ell_t^q\right) \left(\sum_{t=j+1}^{N} s_t^q\right) \left(\sum_{t=1}^{j} \beta_i^t\right) \prod_{j=1}^{N} \left(\sum_{t=j+1}^{N} \beta_i^t\right)
\]

\[
= \prod_{j=1}^{N} \left(2^{s_j^q - \sum_{t=1}^{j} \beta_i^t} \prod_{k=j+1}^{N} s_k^q - \sum_{t=1}^{j} \beta_i^t\right) \prod_{j=1}^{N} \left(\sum_{t=0}^{j-1} \ell_t^q\right) \prod_{j=1}^{N} \left(\sum_{t=j+1}^{N} \beta_i^t\right)
\]
Replacing in (3.9) the values of \( m_{\tau_{k,\ell}}(\mu - \eta_{q,\alpha}^\beta) \) and \( m_{\pi_{l,\ell_2}}(\eta_{q,\alpha}^\beta) \) given by Lemma 3.5 and \#\( \mathcal{P}_{\beta,\alpha}^q \) by the above expression, we obtain the desired weight multiplicity formula for \( \tau_{k,\ell} \). According to (3.4), the proofs of Theorems 3.1, 3.2 and 3.3 are complete.

3.3. Computational comparison. We now include a non-serious computational comparison between the weight multiplicity formulas in Theorems 3.1, 3.2 and 3.3 and Freudenthal’s formula (see for instance [Hu §22.3]). We use the open-source mathematical software Sage and its algebraic combinatorics features developed by the Sage-Combinat community [SCo], which has implemented Freudenthal’s formulas. The source code containing the bivariate algorithm can be found in the public project [PSp] available in CoCalc. (To see the corresponding hyperlink go to the electronic version of this article.)

The word ‘non-serious’ in the previous paragraph has been added for several reasons that we now explain. The formulas proved above have been implemented in Sage by the first named author, who lacks computer programming skills. Thus, their implementations are done poorly and inefficiently. On the contrary, the Sage-Combinat community programmed Freudenthal’s formula in Sage in a very efficient way. Furthermore, the calculations have been made using an old version of Sage [Sa] and a slow computer.

The implementation of Freudenthal’s formula in Sage, called Freudenthal algorithm in the sequel, returns all the weights with their corresponding multiplicities. We suspect that this tactic is due to a matter of efficiency since Freudenthal’s formula is defined recursively. On the other hand, Theorem 3.1, 3.2 and 3.3 compute the multiplicity of a single weight. Thus, in order to make a fair comparison between them, the bivariate algorithm will also determine the set of weights \( \pi_{k_1+l_2} \). To this end, we first find a subset of \( \mathbb{Z}^n \) containing the set of weights of \( \pi_{k_1+l_2} \), namely, \( \{ \mu \in \mathbb{Z}^n : \|\mu\|_1 \leq k + l \} \). Here is a summary of the algorithm.

**Algorithm 3.7** (Bivariate algorithm).

**INPUT:** \( g \) a classical complex Lie algebra of type \( B_n \) or \( C_n \) with \( n \geq 2 \), or \( D_n \) with \( n \geq 3 \), and \( k, l \) non-negative integers.

**OUTPUT:** the sequence of pairs \([\mu, m_{\pi_{k_1+l_2}}(\mu)]\), where \( \mu \) runs over every weight of the representation \( \pi_{k_1+l_2} \) of \( g \) and \( m_{\pi_{k_1+l_2}}(\mu) \) is its multiplicity.

(i) Initialize \( S \) as an empty list.

(ii) Determine the set \( P \) of vectors \( \mu = (a_1, \ldots, a_n) \in \mathbb{Z}^n \) such that \( \|\mu\|_1 \leq k + l \) and \( a_1 \geq a_2 \geq \cdots \geq a_n \geq 0 \).

(iii) Run over all elements \( \mu \) in \( P \).

(iv) Compute \( m_{\pi_{k_1+l_2}}(\mu) \) by Theorems 3.1, 3.3.

(v) In case \( m_{\pi_{k_1+l_2}}(\mu) > 0 \), determine the orbit of \( \mu \) by the group \( W_n \simeq \text{Sym}(n) \times \{\pm 1\}^n \), which acts by permutations and multiplication by \( \pm 1 \) on its entries.

(vi) For each \( \nu \) in the above orbit, add in \( S \) the entry \([\nu, m_{\pi_{k_1+l_2}}(\mu)]\).

(vii) Return \( S \).

**Remark 3.8.** Notice that the set of dominant weights for \( \pi_{k_1+l_2} \) is included in \( P \) introduced in (ii) when \( g \) is of type \( B_n \) or \( C_n \). Although this fact is not true for \( g \) of type \( D_n \), each remaining element has the form \( \bar{\mu} := (a_1, \ldots, a_{n-1}, -a_n) \) for some \( \mu = (a_1, \ldots, a_n) \) in \( P \) with \( a_n > 0 \), and it satisfies \( m_{\pi_{k_1+l_2}}(\bar{\mu}) = m_{\pi_{k_1+l_2}}(\mu) \) since \( n \geq 3 \). Consequently, step (v) obtains all the weights of \( \pi_{k_1+l_2} \) when \( g \) is of type \( B_n \) or \( C_n \) for \( n \geq 2 \) and \( D_n \) for \( n \geq 3 \). Likewise, the group \( W_n \) introduced in (v) coincides with the Weyl group when \( g \) is of type \( B_n \) or \( C_n \). For \( g \) of type \( D_n \) and \( n \geq 3 \), the Weyl group is isomorphic to \( \text{Sym}(n) \times \{\pm 1\}^{n-1} \), thus it is strictly included in \( W_n \). This fact is consistent with the previous comment on the set of dominant weights that is not contained in \( P \).
Table 1 displays the times (in seconds) required by both, the bivariate and Freudenthal, algorithms for different choices of \( n, k \) and \( l \). Let us introduce the notation \( B(X_n, k, l) \) for the time required by our implementation in Sage of the bivariate algorithm for \( g \) of type \( X_n \) (= \( B_n, C_n \) or \( D_n \)) and the irreducible representation of \( g \) having highest weight \( k \epsilon_1 + l \epsilon_2 \). Similarly, write \( F(X_n, k, l) \) for the corresponding required time for the implementation in Sage of Freudenthal algorithm. This abuse of notation (the numbers are periods of time not uniquely determined) will be advantageous to express the numerical conclusions.

We now indicate some conclusions evidenced by the numerical experiments. It is clear that \( B(X_n, k, l) \) is much smaller than \( F(X_n, k, l) \) for coherent small values of \( n, k \) and \( l \). Furthermore, the function \( n \mapsto B(X_n, k, l)/F(X_n, k, l) \) seems to be increasing for any fixed choice of \( X, k \) and \( l \). Moreover, for \( n \) big enough, one would have \( B(X_n, k, l) < F(X_n, k, l) \).

On one hand, we see that \( F(D_n, k, l) < F(C_n, k, l) < F(B_n, k, l) \) and the gaps among them increase when \( n \) grows. The reason is that Freudenthal’s formula depends heavily on the root system associated to \( g \), which is simpler for type \( D_n \) and more complicated for type \( B_n \). On the other hand, \( B(C_n, k, l) \) and \( B(D_n, k, l) \) look similar and \( B(B_n, k, l) \) larger. In this case, the reason is the number of weights. Roughly speaking, the set of weights of \( \pi_{k \epsilon_1 + l \epsilon_2} \) is almost equal to \( \{ \mu \in \mathbb{Z}^n : ||\mu||_1 = k + l, ||\mu||_1 \equiv k + l \ (\text{mod } 2) \} \) for types \( C_n \) and \( D_n \) and to \( \{ \mu \in \mathbb{Z}^n : ||\mu||_1 \leq k + l \} \) for types \( B_n \). In fact, this is a consequence of \( ||\alpha||_1 = 2 \) for every root \( \alpha \) in types \( C_n \) and \( D_n \) and \( ||\alpha||_1 \in \{1, 2\} \) for every root \( \alpha \) in type \( B_n \). Summing up, the bivariate algorithm is not sensible to the number of roots in the corresponding root system, but it is sensible to the one-norm of the roots.

Throughout this paragraph fix a type \( X_n \). The times required by both algorithms depend on \( k + l \). In fact, the set of weights of \( \pi_{k \epsilon_1 + l \epsilon_2} \) does not vary considerably among the different choices of \( k \) and \( l \) with \( k + l \) fixed. Likewise, Freudenthal’s formula is slightly faster when \( l \) grows since the size of the set of weights decreases. However, the bivariate algorithm strongly depends on \( l \). Indeed, as this algorithm involves partitions of all non-negative integers less than or equal to \( l \), its speed reduces when \( l \) increases. In conclusion, fixing the value \( m = k + l \), the function \( l \mapsto B(X_n, m - l, l)/F(X_n, m - l, l) \) attains its minimum when \( l \) is as large as possible, that is, when \( l = k \) or \( l = k - 1 \) according to the parity of \( k + l \). This situation is exemplified in Table 2.

The authors believe that the weight multiplicity formulas in Theorems 3.1–3.3 could be implemented on new versions of Sage. Bivariate representations are a non-trivial class of irreducible representations, which frequently appear on users’ calculations. Not only the time required by the bivariate algorithm for \( n \) large enough is reduced, but there is also a great advantage in the possibility of calculating the multiplicity of a single weight in a very short period of time. For example, when \( g \) is of type \( D_5 \), \( k = 20 \) and \( l = 6 \), its implemented program in Sage takes only between 0.40 and 0.65 seconds for each single weight \( \mu \). Furthermore, the efficiency of the algorithm improves significantly when it returns only the multiplicities of dominant weights (i.e. step (v) is omitted in Algorithm 3.3), which is in general what users really need. This can be appreciated in the fourth column of Table 1 denoted by \( D_n(*) \). There, we list the times required by this simplified version of the bivariate algorithm for \( g \) of type \( D_n \). Of course, the fact that the bivariate algorithm works only for particular simple complex Lie algebras and bivariate representations is a big disadvantage.

### 3.4. Closed explicit weight formulas in particular cases

The weight multiplicity formulas obtained in Theorems 3.1–3.3 are not closed expressions because they involve a sum over partitions of non-negative integers. However, in some particular cases (e.g. small values of \( l \), particular choices of \( \mu \)) it is possible to write out the partitions, and therefore the formulas
Table 1. Computational comparison between bivariate algorithm and Freudenthal’s formula.

| n | k | l | \(B_n\) | \(C_n\) | \(D_n\) | \(D_n(\star)\) | \(B_n\) | \(C_n\) | \(D_n\) |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 5 | 3 | 0.15 | 0.07 | | | 0.32 | 0.13 | |
| 3 | 5 | 3 | 0.26 | 0.14 | 0.15 | 0.13 | 3.88 | 1.87 | 1.27 |
| 4 | 5 | 3 | 0.46 | 0.35 | 0.28 | 0.19 | 32.58 | 14.90 | 12.05 |
| 5 | 5 | 3 | 0.99 | 0.67 | 0.62 | 0.23 | 187.43 | 94.08 | 79.82 |
| 6 | 5 | 3 | 2.82 | 1.89 | 1.78 | 0.24 | 876.17 | 527.69 | 451.43 |
| 7 | 5 | 3 | 6.94 | 5.22 | 4.74 | 0.27 | 3436.25 | 1898.23 | 1961.54 |
| 8 | 5 | 3 | 17.77 | 14.11 | 12.59 | 0.36 | | | |
| 9 | 5 | 3 | 43.23 | 35.47 | 32.11 | 0.51 | | | |
| 10 | 5 | 3 | 97.55 | 87.67 | 78.66 | 0.84 | | | |
| 2 | 10 | 3 | 0.29 | 0.13 | | | 1.92 | 0.48 | |
| 3 | 10 | 3 | 0.82 | 0.49 | 0.45 | 0.37 | 23.68 | 10.23 | 7.85 |
| 4 | 10 | 3 | 2.09 | 1.16 | 1.22 | 0.58 | 291.63 | 130.93 | 108.61 |
| 5 | 10 | 3 | 8.30 | 5.06 | 4.84 | 0.80 | 2630.09 | 1193.45 | 1028.42 |
| 6 | 10 | 3 | 38.42 | 24.86 | 23.90 | 1.16 | | | |
| 7 | 10 | 3 | 183.73 | 146.06 | 126.82 | 1.96 | | | |
| 2 | 50 | 3 | 3.47 | 1.79 | | | 78.13 | 28.54 | |
| 3 | 50 | 3 | 36.40 | 17.62 | 17.50 | 13.50 | 5146.69 | 2108.21 | 1578.78 |
| 4 | 50 | 3 | 472.14 | 325.76 | 267.25 | 58.00 | | | |
| 2 | 6 | 6 | 2.20 | 1.32 | | | 0.50 | 0.19 | |
| 3 | 6 | 6 | 9.28 | 5.35 | 5.23 | 5.21 | 11.90 | 4.74 | 3.53 |
| 4 | 6 | 6 | 19.29 | 11.19 | 11.44 | 11.09 | 157.23 | 67.93 | 54.79 |
| 5 | 6 | 6 | 30.81 | 18.05 | 18.04 | 15.78 | 1443.41 | 663.86 | 553.57 |
| 6 | 6 | 6 | 53.76 | 32.65 | 32.64 | 19.50 | | | |
| 2 | 10 | 6 | 3.58 | 2.04 | | | 1.66 | 0.62 | |
| 3 | 10 | 6 | 18.14 | 9.86 | 9.91 | 9.77 | 43.50 | 18.34 | 13.59 |
| 4 | 10 | 6 | 44.69 | 25.20 | 24.96 | 23.98 | 695.71 | 298.50 | 243.70 |
| 5 | 10 | 6 | 87.71 | 49.55 | 49.36 | 38.89 | 8114.00 | 3571.44 | 2966.84 |
| 6 | 10 | 6 | 235.76 | 158.26 | 133.51 | 52.88 | | | |
| 2 | 20 | 6 | 8.77 | 4.61 | | | 8.41 | 3.16 | |
| 3 | 20 | 6 | 63.71 | 33.25 | 33.16 | 32.61 | 312.83 | 129.92 | 98.02 |
| 4 | 20 | 6 | 216.46 | 115.58 | 117.99 | 109.08 | 7486.63 | 3199.29 | 2620.10 |
| 5 | 20 | 6 | 654.96 | 390.98 | 393.92 | 220.11 | | | |
| 2 | 15 | 9 | 40.04 | 22.33 | | | 4.93 | 1.81 | |
| 3 | 15 | 9 | 390.59 | 209.87 | 209.12 | 208.37 | 191.24 | 78.85 | 59.11 |
| 4 | 15 | 9 | 1594.63 | 865.10 | 853.83 | 851.50 | 4710.03 | 1908.34 | 1642.50 |
| 5 | 15 | 9 | 3794.15 | 2112.98 | 2051.99 | 1962.57 | 71389.97 | 32013.22 | 28179.33 |
| 2 | 50 | 9 | 231.18 | 116.93 | | | 96.16 | 35.20 | |
| 3 | 50 | 9 | 4800.55 | 2423.05 | 2553.71 | 2492.15 | 7851.47 | 3117.85 | 2346.84 |

Each column shows, for the corresponding algorithm and type X, the required time for returning the set of weights with multiplicities of the representation \(\pi_{k+1}+l\epsilon_2\) of \(g\) of type \(X_n\) according to the row. The column \(D_n(\star)\) refers to the version of the bivariate algorithm returning only the dominant weights.
become closed expressions. For example, if \( l = 0 \) then the formulas reduce to the closed explicit expressions in Lemma 3.5.

When \( l = 1 \), only sums over the set of partitions of 0 or 1 are involved. These sets have exactly one element, so the sums disappear. For example, when \( g \) is of type \( D_n \), we get

\[
(3.10) \quad m_{\pi_{k+1+c}}(\mu) = D_n(1, r(\mu), \ell_0(\mu)) - D_n(0, r(\mu)) - D_n(0, r(\mu) - 1)
\]

\[
= \sum_{|b| = 0} \sum_{|a| = 0} (r - 1 + \alpha_1 + n - 2) \binom{\beta_1}{\alpha_1} (n - \ell_0(\mu)) (\ell_0(\mu))
\]

\[
- \binom{r(\mu) + n - 2}{n - 2} - \binom{r(\mu) - 1 + n - 2}{n - 2}
\]

for every \( \mu \in \mathbb{Z}^n \) satisfying that \( r(\mu) = (k + 1 - ||\mu||_1)/2 \) is a non-negative integer. Notice that this formula is a particular case of [LR17, Thm. 4.1].

Similarly, when \( l = 2 \) there are only sums over the set of partitions of \( N \) for \( N = 0, 1, 2 \). Since \( 2 = 2 \) and \( 2 = 1 + 1 \) are the only partitions of \( 2 \), the corresponding sum splits in two. We now state the multiplicity formula for \( l = 2 \) and type \( D_n \). We pick type \( D_n \) for citing purposes.

**Corollary 3.9.** Let \( g = \mathfrak{so}(2n, \mathbb{C}) \) for \( n \geq 3 \), let \( k \geq 2 \) integer and let \( \mu \in \mathbb{Z}^n \). If \( r(\mu) := (k + 2 - ||\mu||_1)/2 \) is a non-negative integer, then

\[
m_{\pi_{k+1+2c}}(\mu) = \frac{r(\mu) + n - 4}{n - 2} \binom{2\ell_0(\mu)(n - 1)}{2}\left(\frac{n - \ell_0(\mu)}{2}\right)
\]

\[
+ \binom{r(\mu) + n - 3}{n - 2} \left(\frac{2\ell_0(\mu)(n - \ell_0(\mu)) + \ell_1(\mu) - n}{2}\right)
\]

\[
+ \binom{r(\mu) + n - 2}{n - 2} \left(\frac{n - \ell_0(\mu)}{2} - \ell_1(\mu)\right),
\]

and \( m_{\pi_{k+1+2c}}(\mu) = 0 \) otherwise.

Furthermore, we can obtain a closed explicit multiplicity formula for the weight \( \mu = 0 \) in the representation \( \pi_{k+1+2c} \) of \( g \). We next state the formulas for types \( B_n \), \( C_n \) and \( D_n \), but we prove it only for the case \( D_n \), since types \( B_n \) and \( C_n \) follow in a similar way.

**Corollary 3.10.** (Type \( D_n \)) Let \( g = \mathfrak{so}(2n, \mathbb{C}) \) for some \( n \geq 3 \) and let \( k \geq l \geq 0 \) integers. We have that \( m_{\pi_{k+1+l}}(0) = 0 \) if \( k + l \) is odd. Moreover, if \( k + l \) is even, then

\[
m_{\pi_{k+1+l}}(0) = 2 \sum_{0 \leq N \leq l} (-1)^{N+l} R(n, k, l, N) \left(\frac{[(l - N)/2] + n - 2}{n - 2}\right)
\]

\[
\left(\frac{[(k - N + 1)/2] + n - 2}{n - 2}\right) \sum_{t=0}^{n} \binom{n}{t} \left(\frac{N - t + n - 1}{n - 1}\right),
\]
where

\[ R(n, k, l, N) = \begin{cases} 
\frac{l - N + n - 2}{l - N + 2n - 4} & \text{if } N \equiv l \pmod{2}, \\
\frac{k + 1 - N + n - 2}{k + 1 - N + 2n - 4} & \text{if } N \equiv l + 1 \pmod{2}.
\end{cases} \]

Proof. The asserted formula can be obtained by Theorem 3.3. However, we will prove it in a simplified way, by following the proof of Theorem 3.3. The reason is that the partition in (3.8) of the set of weights of \( \pi_{\ell_1} \) is (unnecessarily) too fine for \( \mu = 0 \). By (3.4), we have that

(3.11) \[ m_{\pi_{k+1}+l_2}(0) = m_{\tau_{k,l}}(0) - m_{\tau_{k+1,l-1}}(0) - m_{\tau_{k-1,l-1}}(0) + m_{\tau_{k,l-2}}(0). \]

As before, for arbitrary \( k \geq l \geq 0 \) integers, it holds

\[ m_{\tau_{k,l}}(0) = \sum_{\eta} m_{\pi_{k\ell_1}}(-\eta) m_{\pi_{\ell_1}}(\eta), \]

where the sum is restricted to the weights of \( \pi_{\ell_1} \). From Lemma 3.5, we see that \( \eta \) is a weight of \( \pi_{\ell_1} \) if and only if \( l - \|\eta\|_1 \in 2\mathbb{N}_0 \). For such a weight \( \eta \), \( m_{\pi_{k\ell_1}}(-\eta) = 0 \) unless \( 2\mathbb{N}_0 \ni k - \|\eta\|_1 = (k - l) + (l - \|\eta\|_1) \), equivalently \( k - l \in 2\mathbb{N}_0 \). We conclude that \( m_{\tau_{k,l}}(0) = 0 \) if \( k + l \) is odd. Moreover, \( m_{\pi_{k+1}+l_2}(0) = 0 \) if \( k + l \) is odd by (3.11).

We now proceed to compute \( m_{\tau_{k,l}}(0) \) for arbitrary \( k \geq l \geq 0 \) integers satisfying that \( k + l \) is even. Fix \( N \in \mathbb{N}_0 \) such that \( l - N \in 2\mathbb{N}_0 \). For each \( \eta \in \mathbb{Z}^n \) with \( \|\eta\|_1 = N \) we know that \( m_{\pi_{k\ell_1}}(-\eta) \) and \( m_{\pi_{\ell_1}}(\eta) \) are constant, independent of the choice of \( \eta \). Hence,

(3.12) \[ m_{\tau_{k,l}}(0) = \sum_{0 \leq N \leq l, \ N \equiv l \pmod{2}} \left( \frac{(l - N)/2 + n - 2}{n - 2} \right) \left( \frac{(k - N)/2 + n - 2}{n - 2} \right) \# \{ \eta \in \mathbb{Z}^n : \|\eta\|_1 = N \}. \]

It is well known (see for instance [BR, §2.5]) that \( \# \{ \eta \in \mathbb{Z}^n : \|\eta\|_1 = N \} = \sum_{t=0}^{n} \binom{n}{t} \binom{N-t+n-1}{n-1}. \)

Thus, by replacing (3.12) in (3.11) one obtains the desired formula.

Corollary 3.11. (Type C\(_n\)) Let \( g = sp(n, \mathbb{C}) \) for some \( n \geq 2 \) and let \( k \geq l \geq 0 \) integers. We have that \( m_{\pi_{k\ell_1}+l_2}(0) = 0 \) if \( k + l \) is odd. Moreover, if \( k + l \) is even then

\[ m_{\pi_{k+1}+l_2}(0) = 2 \sum_{0 \leq N \leq l} (-1)^{N+l} R(n + 1, k, l, N) \left( \frac{[l - N/2 + n - 1]}{n - 1} \right) \left( \frac{[k - N + 1/2 + n - 1]}{n - 1} \right) \sum_{t=0}^{n} \binom{n}{t} \binom{N - t + n - 1}{n - 1}, \]

where \( R(n, k, l, N) \) is as in Corollary 3.10.

Corollary 3.12. (Type B\(_n\)) Let \( g = so(2n + 1, \mathbb{C}) \) for some \( n \geq 2 \) and let \( k \geq l \geq 0 \) integers. Then

\[ m_{\pi_{k\ell_1}+l_2}(0) = \sum_{0 \leq N \leq l} (-1)^{N+l} S(n, k, l, N) \left( \frac{[l - N/2 + n - 1]}{n - 1} \right) \left( \frac{[k + 1 - N/2 + n - 1]}{n - 1} \right) \sum_{t=0}^{n} \binom{n}{t} \binom{N - t + n - 1}{n - 1}, \]
where
\[
S(n, k, l, N) = \begin{cases} 
1 - \frac{[(l - N)/2][(k + 1 - N)/2]}{[(l - N)/2] + n - 1][(k + 1 - N)/2] + n - 1} & \text{if } k + l \text{ is even,} \\
\frac{[(k + 1 - N)/2]}{[(k + 1 - N)/2] + n - 1} - \frac{[(l - N)/2]}{[(l - N)/2] + n - 1} & \text{if } k + l \text{ is odd.}
\end{cases}
\]

3.5. Remarks. We end this section with a few remarks.

Remark 3.13. The weight multiplicity formula for type $D_n$ in Theorem 3.3 also holds when $n = 2$ with $\pi_{k_{e_1} \pm l_{e_2}}$ replaced by $\pi_{k_{e_1} \pm l_{e_2}}$. It is important to note that $\mathfrak{so}(4, \mathbb{C})$ (type $D_2$) is not simple. Indeed, $\mathfrak{so}(4, \mathbb{C}) \simeq \mathfrak{sl}(2, \mathbb{C}) \oplus \mathfrak{sl}(2, \mathbb{C})$ or $D_2 = A_1 \oplus A_1$. Hence, a weight multiplicity formula for the representations $\pi_{k_{e_1} \pm l_{e_2}}$ with $k \geq l \geq 0$ of $\mathfrak{so}(4, \mathbb{C})$ can be obtained using this fact.

Remark 3.14. Maddox in [Ma14] determined a weight multiplicity formula for any bivariate representation for $\mathfrak{g}$ of type $C_n$. Her expression ([Ma14, Thm. 4.3]) looks more elegant than the one in Theorem 3.2. However, it includes a sum over ordered partitions of $r(\mu)$ of length $n$ and another sum over the subsets of a set of $2n$ elements. In conclusion, her shorter formula hides in the mentioned sums the involved terms appearing in the expression given in Theorem 3.2. Furthermore, the neat dependence condition in Theorem 1.1 does not follow immediately from [Ma14, Thm. 4.3].

We now compare Maddox’s method with ours. Both employ the expression in Lemma 3.6 for an irreducible representation as sum of tensor products in the virtual ring of representations. The significant difference arises in the calculation of the weight multiplicity in a tensor product. Roughly speaking, the proofs of Theorems 3.1–3.3 use the identity (3.6) and then a convenient partition of the set of weights of the small component in the tensor product. On the other hand, Maddox makes use of $\tau_{k, l} = \pi_{k_{e_1}} \otimes \pi_{l_{e_2}} \simeq \text{Sym}^k(\mathbb{C}^{2n}) \otimes \text{Sym}^l(\mathbb{C}^{2n})$ for $\mathfrak{g}$ of type $C_n$, and counts the weight vectors in terms of a function which has a combinatorial expression.

Remark 3.15. There are in the literature several algorithms to compute weight multiplicities. The one based on Freudenthal’s formula is the most classical and is still used for several computer program (e.g. Sage [Sa]). Nowadays, there exist faster algorithms. A possible time comparison with any of them would require an implementation on Sage, which would be unfair because of the poor computer programming skills of the authors.

Among the mentioned faster algorithms, it is distinguished the one by W. Baldoni and M. Vergne [BV18] (see also [BBCV06, BV15, Co05] for related results), which is based on symbolic computations of Kostant partition functions. See also [CDW12, Sc12, Ca17] for recent different approaches.

Remark 3.16. This interesting remark about the behavior of $m_{\pi_{k_{e_1} \pm l_{e_2}}}(\mu)$ as a function on $k$ and $l$ was pointed out by the referee. For simplicity, we take $\mu = 0$, we fix $l$ a non-negative integer, and we consider $\mathfrak{g}$ a classical Lie algebra of type $D_n$ for some $n \geq 3$, although the general case is very similar. Corollary 3.10 implies that $k \mapsto m_{\pi_{k_{e_1} \pm l_{e_2}}}(0)$ is a quasi-polynomial in the variable $k \geq l$ whose degree does not depend on $l$. In fact, its degree coincides with the degree of the polynomial $k \mapsto m_{\pi_{2k_{e_1}}}(0) = \binom{k+n-2}{n-2}$ (i.e. when $l = 0$), which is equal to $n - 2$.

An interesting problem, also suggested by the referee, is to understand the behavior of the function $l \mapsto m_{\pi_{(l+h)_{e_1} \pm l_{e_2}}}(0)$, for some $h$ fixed. This does not seem to be computable from Corollary 3.10.

Remark 3.17. In [LR17, Section 7] there is a detailed account of some applications of weight multiplicity formulas in spectral geometry (see [LMR16, BL17, La16, La18]). These expressions for the weight multiplicities are used to determine explicitly the spectra of certain
natural differential operators on a manifold (or a good orbifold) of the form $\Gamma \backslash G/K$, where $G$ is a semisimple compact Lie group, $K$ is a closed subgroup of $G$ and $\Gamma$ is a finite subgroup of the maximal torus $T$ of $G$.

We next specify some cases where the formulas obtained in this article could be applied. When $G = \text{Sp}(n)$ and $K = \text{Sp}(n-1) \times \text{Sp}(1)$, the spherical representations associated to the Gelfand pair $(G, K)$ (i.e. the set of irreducible representations of $G$ containing non-zero vectors fixed by $K$) have highest weight of the form $k(\varepsilon_1 + \varepsilon_2)$ for $k \geq 0$. Consequently, Theorem 3.2 may be applied to determine the spectrum of the Laplace–Beltrami operator acting on functions on spaces covered by the $n$-dimensional quaternionic projective space $\text{Sp}(n)/\text{Sp}(n-1) \times \text{Sp}(1)$ with abelian fundamental group.

When $G = \text{SO}(m)$ and $K = \text{SO}(m-2) \times \text{SO}(2)$, the corresponding spherical representations for $(G, K)$ have highest weight of the form $lk\varepsilon_1 + l\varepsilon_2$ for $k \geq l \geq 0$. Thus, according to $m$ is odd or even, Theorems 3.1 or 3.3 could be applied to the same purpose as above, for spaces covered by the 2-Grassmannian space $G/K$ with abelian fundamental group.

In a slightly different way, we now consider $n \geq 3$, $G = \text{SO}(2n)$, $K = \text{SO}(2n-1)$ and more general natural differential operators. An irreducible representation $\tau$ of $K$ induces a natural $G$-homogeneous complex vector bundle $E_\tau$ on $G/K$. There is an associated natural differential operator $\Delta_\tau$ acting on smooth sections of $E_\tau$, which induces the differential operator $\Delta_{\tau,\Gamma}$ acting on smooth sections of $\Gamma \backslash E_\tau$, that is, $\Gamma$-invariant smooth sections of $E_\tau$. We now fix $\tau = \tau_{b_1}$, the irreducible representation of $K$ with highest weight $b_1$. The corresponding $\tau_{b_1}$-spherical representations of $(G, K, \tau_{b_1})$ (i.e. the set of $\pi \in \hat{G}$ such that $\text{Hom}_K(\tau_{b_1}, \pi|_K) \neq 0$) is equal to $\{\pi_{k\varepsilon_1 + l\varepsilon_2} : k \geq b, l \geq 0\}$. Consequently, Theorem 3.3 might be used to determine the spectrum of $\Delta_{\tau_{b_1},\Gamma}$ for $\Gamma$ a finite subgroup of the maximal torus of $G$. An analogous process can be done in the case $G = \text{SO}(2n-1)$ and $K = \text{SO}(2n-2)$.

4. Type $A_n$

Consider in $\mathfrak{g} = \mathfrak{sl}(n+1, \mathbb{C})$ the Cartan subalgebra

$$\mathfrak{h} = \{\text{diag}(\theta_1, \ldots, \theta_{n+1}) : \theta_i \in \mathbb{C} \forall i, \sum_{i=1}^{n+1} \theta_i = 0\}.$$ 

Set $\varepsilon_i(\text{diag}(\theta_1, \ldots, \theta_{n+1})) = \theta_i$ for each $1 \leq i \leq n+1$. We will use the conventions of [FH, Lecture 15], that is, we correspondingly write

$$\mathfrak{h}^* = \bigoplus_{i=1}^{n+1} \mathbb{C} \varepsilon_i / \langle \sum_{i=1}^{n+1} \varepsilon_i = 0 \rangle,$$

and we write $\varepsilon_i$ for its image in $\mathfrak{h}^*$. Consequently, the set of positive roots is given by $\Sigma^+(\mathfrak{g}, \mathfrak{h}) := \{\varepsilon_i - \varepsilon_j : 1 \leq i < j \leq n+1\}$ and the weight lattice is $P(\mathfrak{g}) := \bigoplus_{i=1}^{n+1} \mathbb{Z} \varepsilon_i / \langle \sum_{i=1}^{n+1} \varepsilon_i = 0 \rangle$. Two weights $\mu = \sum_{i=1}^{n+1} b_i \varepsilon_i$ and $\nu = \sum_{i=1}^{n+1} c_i \varepsilon_i$ in $P(\mathfrak{g})$ coincide if and only if $b_i - c_i$ is constant, independent of $i$.

A weight $\lambda = \sum_{i=1}^{n+1} a_i \varepsilon_i$ in $P(\mathfrak{g})$ is dominant if and only if $a_1 \geq a_2 \geq \cdots \geq a_{n+1}$. By the Highest Weight Theorem, the irreducible representations of $\mathfrak{g}$ are in correspondence with dominant weights. We denote by $\pi_\lambda$ the irreducible representation with highest weight $\lambda$, which will be always written as $\lambda = \sum_{i=1}^{n+1} a_i \varepsilon_i$ with $a_{n+1} = 0$. Thus, the irreducible representations of $G$ are in correspondence with elements in the set

$$P^{++}(\mathfrak{g}) := \left\{ \sum_{i=1}^{n} a_i \varepsilon_i : a_i \in \mathbb{Z} \forall i, \ a_1 \geq a_2 \geq \cdots \geq a_n \geq 0 \right\}.$$
The fundamental weights are given by \( \omega_p = \epsilon_1 + \cdots + \epsilon_p \) for each \( 1 \leq p \leq n \). Thus, any integer combination between \( \omega_1 \) and \( \omega_2 \) has the form \( k\epsilon_1 + l\epsilon_2 \) for some integers \( k \geq l \geq 0 \).

For \( \lambda = \sum_{i=1}^{n} a_i \epsilon_i \in P^+(g) \), any weight \( \mu \) of \( \pi_\lambda \) (i.e. the multiplicity of \( \mu \) in \( \pi_\lambda \) is non-zero) can be written as \( \mu = \sum_{i=1}^{n+1} b_i \epsilon_i \) for some \( b_1, \ldots, b_{n+1} \in \mathbb{N}_0 \) satisfying \( \sum_{i=1}^{n+1} b_i = \sum_{i=1}^{n} a_i \).

Indeed, every weight in \( \pi_\lambda \) is a difference between \( \lambda \) and a sum of positive roots.

Let \( \lambda \) and \( \mu \) as in the previous paragraph. It is well known that (see for instance [FH (A.19)]) the multiplicity of \( \mu \) in \( \pi_\lambda \) is given by the Kostka number \( K_{\lambda,\mu} \): the number of semistandard tableaux on the Young diagram associated to \( \lambda \) (i.e. a diagram with \( a_i \) boxes in the \( i \)th row, with the rows of boxes lined up on the left) of type \( \mu \). More precisely, \( K_{\lambda,\mu} \) is the number of ways one can fill the boxes of the Young diagram associated to \( \lambda \) with \( b_1 \) 1’s, \( b_2 \) 2’s, up to \( b_{n+1} \) \( (n+1) \)’s, in such a way that the entries in each row are non-decreasing, and those in each column are strictly increasing.

The next lemma will be needed in the proof of the main result of this section.

**Lemma 4.1.** Let \( g \) be a classical Lie algebra of type \( A_n \). For integers \( k \geq l \geq 0 \) write \( \tau_{k,l} = \pi_{k \epsilon_1} \otimes \pi_{l \epsilon_1} \). Then, in the virtual ring of representations, we have that

\[
\pi_{k \epsilon_1 + l \epsilon_2} \simeq \tau_{k,l} - \tau_{k+1,l-1}.
\]

**Proof.** The well-known fusion rule (see for instance [FH, Prop. 15.25])

\[
\tau_{k,l} = \pi_{k \epsilon_1} \otimes \pi_{l \epsilon_1} \simeq \bigoplus_{p=0}^l \pi_{(k+p)\epsilon_1 + (l-p)\epsilon_2}
\]

implies

\[
\tau_{k,l} - \tau_{k+1,l-1} = \sum_{p=0}^l \pi_{(k+p)\epsilon_1 + (l-p)\epsilon_2} - \sum_{p=1}^l \pi_{(k+p)\epsilon_1 + (l-p)\epsilon_2} = \pi_{k \epsilon_1 + l \epsilon_2},
\]

and the lemma follows. \( \square \)

We now want to calculate the weight multiplicities of the representation with highest weight a non-negative integer combination of the first two fundamental weights. The following multiplicity formula is probably already known, but it is included here for completeness.

**Theorem 4.2.** (Type \( A_n \)) Let \( g = \mathfrak{sl}(n+1, \mathbb{C}) \) for some \( n \geq 2 \) and let \( k \geq l \geq 0 \) integers. Let \( \mu = \sum_{i=1}^{n+1} a_i \epsilon_i \in P(g) \) with \( a_i \in \mathbb{N}_0 \) for all \( i \) and \( \sum_{i=1}^{n+1} a_i = k + l \). If \( a_i \leq k \) for all \( i \), then

\[
m_{\pi_{k \epsilon_1 + l \epsilon_2}}(\mu) = \sum_{q \in \mathcal{Q}_{n+1}(l)} \prod_{j=1}^l \left( n + 1 - \sum_{t=0}^{j-1} \ell_t(\mu) - \sum_{i=j+1}^{l} s^q_i \right)
\]

\[
- \sum_{q' \in \mathcal{Q}_{n+1}(l-1)} \prod_{j=1}^{l-1} \left( n + 1 - \sum_{t=0}^{j-1} \ell_t(\mu) - \sum_{i=j+1}^{l-1} s^{q'}_i \right),
\]

and \( m_{\pi_{k \epsilon_1 + l \epsilon_2}}(\mu) = 0 \) otherwise, where \( \ell_t(\mu) = \# \{ i : 1 \leq i \leq n + 1, a_i = t \} \),

\[
\mathcal{Q}_{n+1}(N) = \{ q = (q_1, q_2, \ldots, q_{n+1}) \in \mathbb{Z}^{n+1} : q_1 \geq q_2 \geq \cdots \geq q_{n+1} \geq 0, \sum_{i=1}^{n+1} q_i = N \},
\]

and \( s^q_i := \# \{ i : 1 \leq i \leq n + 1, q_i = j \} \) for \( q \in \mathcal{Q}_{n+1}(N) \) and \( 1 \leq j \leq N \).

**Proof.** Since \( \tau_{k,l} = \pi_{k \epsilon_1} \otimes \pi_{l \epsilon_1} \), we have that

\[
m_{\tau_{k,l}}(\mu) = \sum_{\eta} m_{\pi_{k \epsilon_1}}(\mu - \eta) m_{\pi_{l \epsilon_1}}(\eta),
\]

and
where the sum is restricted to the weights of $\pi_{\ell_1}$. For $h$ any positive integer, the Young diagram associated to $\pi_{k\varepsilon_1}$ has only one row, of length $h$. Thus, the set of weights of $\pi_{h\varepsilon_1}$ is given by elements of the form $\nu = \sum_{i=1}^{n+1} c_i \varepsilon_i$ with $c_1, \ldots, c_{n+1} \in \mathbb{N}_0$ and $\sum_{i=1}^{n+1} c_i = h$, and all of them have multiplicity 1. Consequently, $m_{\pi_{k\varepsilon_1}}(\mu)$ is equal to the number of weights $\eta$ of $\pi_{\ell_1}$ satisfying that $\mu - \eta$ is a weight of $\pi_{k\varepsilon_1}$.

Let $q \in Q_{n+1}(l)$. We want to count the number of weights $\eta = \sum_{i=1}^{n+1} b_i \varepsilon_i$ contributing to $\pi_{k\varepsilon_1}$ (i.e. $\eta$ is a weight of $\pi_{\ell_1}$ and $\mu - \eta$ is a weight of $\pi_{k\varepsilon_1}$) satisfying that $s_j^q$ entries of $\eta$ are equal to $j$ for each $1 \leq j \leq l$. Clearly, $\mu - \eta$ is a weight of $\pi_{k\varepsilon_1}$, if and only if $a_i - b_i \geq 0$ for all $1 \leq i \leq n + 1$. Since for each $1 \leq j \leq l$ there are $n + 1 - \sum_{t=0}^{j-1} \ell_t(\mu) a_i$’s greater than $j - 1$, then the required number is

$$(4.2) \quad \left( n + 1 - \sum_{t=0}^{l-1} \ell_t(\mu) \right) \left( n + 1 - \sum_{t=0}^{l-2} \ell_t(\mu) - s_l^q \right) \ldots \left( n + 1 - \ell_0(\mu) - \sum_{j=2}^{l} s_j^q \right).$$

We have shown that $m_{\pi_{k\varepsilon_1}}(\mu)$ is equal to the sum over $q \in Q_{n+1}(l)$ of $(4.2)$. The theorem now follows by Lemma 4.1.

We now state the closed explicit formulas for the particular cases $l = 0$, 1, and 2. When $l = 0$, since $Q_{n+1}(0) = \{(0, \ldots, 0)\}$, Theorem 4.2 immediately implies that every weight as in the hypotheses (i.e. $\mu = \sum_{i=1}^{n+1} b_i \varepsilon_i$ with $b_i \in \mathbb{N}_0$ for all $i$ and $\sum_{i=1}^{n+1} b_i = k$) has multiplicity one. This fact is very well known because the Young diagram associated to $\pi_{k\varepsilon_1}$ has only one row, and consequently, the number of semistandard tableaux on this diagram of type $\mu$ is one.

We now assume $l = 1$. Let $\mu$ again as in the hypotheses of Theorem 4.2. The number of partitions of 1 is obviously one, i.e. $Q_{n+1}(1) = \{(1, 0, \ldots, 0)\}$, thus $m_{\pi_{k\varepsilon_1+\varepsilon_1}}(\mu) = (n+1-\ell_0(\mu)) - 1 = n - \ell_0(\mu)$, where $\ell_0(\mu)$ is the number of zeros coordinates of $\mu$. It is not difficult to check that the number of semistandard tableaux of type $\mu$ is $n - \ell_0(\mu)$.

We conclude the article stating the multiplicity formula for the irreducible representation of $\mathfrak{sl}(n+1, \mathbb{C})$ with highest weight $k\varepsilon_1 + 2\varepsilon_2$. Similarly as above, the proof follows immediately from Theorem 4.2 since it reduces to consider the only two partitions of 2. The reader may try to obtain this formula by counting semistandard tableaux of type $\mu$, and convince his/her self that the difficulty will increase for a higher $l$.

**Corollary 4.3.** Let $g = \mathfrak{sl}(n+1, \mathbb{C})$ for some $n \geq 2$ and let $k \geq 2$ integer. Let $\mu = \sum_{i=1}^{n+1} a_i \varepsilon_i \in P(g)$ with $a_i \in \mathbb{N}_0$ for all $i$ and $\sum_{i=1}^{n+1} a_i = k + 2$. If $a_i \leq k$ for all $i$, then

$$m_{\pi_{k\varepsilon_1+2\varepsilon_2}}(\mu) = \left( n + 1 - \ell_0(\mu) \right) - \ell_1(\mu)$$

and $m_{\pi_{k\varepsilon_1+2\varepsilon_2}}(\mu) = 0$ otherwise, where $\ell_t(\mu) = \# \{ i : 1 \leq i \leq n + 1, a_i = t \}$.

We end this article with an observation pointed out by the referee, in the same spirit of Remark 3.16.

**Remark 4.4.** We consider the ‘weight zero’ in $\pi_{k\varepsilon_1+\varepsilon_2}$, which in our convention is given by $0_{k+l} := \sum_{i=1}^{n+1} \frac{k+l}{n+1} \varepsilon_i$. Clearly, $m_{\pi_{k\varepsilon_1+\varepsilon_2}}(0_{k+l}) = 0$ unless $n + 1$ divides $k + l$. Theorem 4.2 does not give an explicit expression for $m_{\pi_{k\varepsilon_1+\varepsilon_2}}(0_{k+l})$ like in Corollaries 3.10–3.12. However, for $l \geq 0$ fixed, it implies that $m_{\pi_{k\varepsilon_1+\varepsilon_2}}(0_{k+t})$ does not depend on $k$, for $k$ sufficiently large satisfying that $n + 1$ divides $k + l$. Moreover, the function $k \mapsto m_{\pi_{k\varepsilon_1+\varepsilon_2}}(0_{k+t})$ is constant for every $k \in [n + (n + 1)\mathbb{N}_0]$. Indeed, for such $k$, we have that $\frac{k+l}{n+1} \geq l$, thus $\ell_t(0_{k+l}) = 0$ for every $0 \leq t \leq l - 1$. 

Acknowledgments

The authors wish to thank the anonymous referee for pointing out very interesting remarks. This research was partially supported by grants from CONICET, FO NCyT and SeCyT–UNC. The first named author was supported by the Alexander von Humboldt Foundation.

References

[BBCV06] M.W. Baldoni, M. Beck, C. Cochet, M. Vergne. Volume computation for polytopes and partition functions for classical root systems. Discrete Comput. Geom. 35:4 (2006), 551–595. DOI: 10.1007/s00454-006-1234-2.

[BV15] V. Baldoni, M. Vergne. Multiplicity of compact group representations and applications to Kronecker coefficients. arXiv:1506.02472 (2015).

[BV18] V. Baldoni, M. Vergne. Computation of dilated Kronecker coefficients. (With an appendix by M. Walter.) J. Symbolic Comput. 84 (2018), 113–146. DOI: 10.1016/j.jsc.2017.03.005.

[BR] M. Beck, S. Robins. Computing the continuous discretely. Undergrad. Texts Math., Springer New York, 2007. DOI: 10.1007/978-0-387-46112-0.

[BL17] S. Boldt, E.A. Lauret. An explicit formula for the Dirac multiplicities on lens spaces. J. Geom. Anal. 27 (2017), 689–725. DOI: 10.1007/s12220-016-9695-3.

[Ca17] M. Cavallin. An algorithm for computing weight multiplicities in irreducible modules for complex semisimple Lie algebras. J. Algebra 471 (2017), 492–510. DOI: 10.1016/j.jalgebra.2016.08.044.

[CDW12] M. Christandl, B. Doran, M. Walter. Computing multiplicities of Lie group representations. In 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science—FOCS 2012, 639–648, IEEE Computer Soc., Los Alamitos, CA, 2012.

[Co05] C. Cochet. Vector partition function and representation theory. Conference Proceedings on Formal Power Series and Algebraic Combinatorics, Taormina, Italy (2005), 1009–1020.

[FH] W. Fulton, J. Harris. Representation Theory, A first course. Springer-Verlag New York, 2004. DOI: 10.1007/978-1-4612-0979-9.

[Hu] J. Humphreys. Introduction to Lie Algebras and their representations. Grad. Texts in Math. 9. Springer-Verlag New York, 1972. DOI: 10.1007/978-1-4612-6398-2.

[Kn] A.W. Knapp. Lie groups beyond an introduction. Progr. Math. 140. Birkhäuser Boston Inc., 2002.

[KT87] K. Koike, I. Terada. Young-diagrammatic methods for the representation theory of the classical groups of type $B_n$, $C_n$, $D_n$. J. Algebra 107 (1987), 466–511. DOI: 10.1016/0021-8693(87)90099-8.

[La16] E.A. Lauret. Spectra of orbifolds with cyclic fundamental groups. Ann. Global Anal. Geom. 50:1 (2016), 1–28. DOI: 10.1007/s10455-016-9498-0.

[La18] E.A. Lauret. The spectrum on $p$-forms of a lens space. Geom. Dedicata, in press since February 2018. DOI: 10.1007/s10711-018-0322-9.

[LMR16] E.A. Lauret, R.J. Miatello, J.P. Rossetti. Spectra of lens spaces from 1-norm spectra of congruence lattices. Int. Math. Res. Not. IMRN 2016:4 (2016), 1054–1089. DOI: 10.1093/imrn/rnv159.

[LR17] E.A. Lauret, F. Rossi Bertone. Multiplicity formulas for fundamental strings of representations of classical Lie algebras. J. Math. Phys. 58 (2017), 111703. DOI: 10.1063/1.4993851.

[Ma14] J. Maddox. An elementary approach to weight multiplicities in bivariate irreducible representations of $Sp(2r)$. Comm. Algebra 42:9 (2014), 4094–4101. DOI: 10.1080/00927872.2013.804928.

[PSp] Public Sage project Weight multiplicities for bivariate representations in Collaborative Calculation in the Cloud (CoCalc.com). 2017.

[Sa] W.A. Stein et al. Sage Mathematics Software (Version 4.3). The Sage Development Team, 2009, www.sagemath.org.

[SCo] The Sage-Combinat community, Sage-Combinat: enhancing Sage as a toolbox for computer exploration in algebraic combinatorics, http://combinat.sagemath.org, 2008.

[Sc12] W. Schützer. A new character formula for Lie algebras and Lie groups. J. Lie Theory 22:3 (2012), 817–838.

CIEM–FAMAF (CONICET), Universidad Nacional de Córdoba, Medina Allende s/n, Ciudad Universitaria, 5000 Córdoba, Argentina.

E-mail address: elauret@famaf.unc.edu.ar, rossib@famaf.unc.edu.ar