Constructing representations of split semisimple Lie algebras

W. A. de Graaf

School of Computer Science
University of St Andrews
St Andrews Scotland
e-mail: `wdg@dcs.st-and.ac.uk`

Abstract

We describe an algorithm for constructing irreducible representations of split semisimple Lie algebras in characteristic 0. The algorithm calculates a Gröbner basis of a certain left ideal in a universal enveloping algebra. It is shown that this algorithm runs in polynomial time if the Lie algebra is fixed. At the end of the paper practical experiences with an implementation of the algorithm in GAP4 are discussed.

1 Introduction

In this paper we deal with "split" semisimple Lie algebras of characteristic 0. These are by definition semisimple Lie algebras L with a Cartan subalgebra H such that the field of definition contains the eigenvalues of all transformations $\mathrm{ad}\, h$ for $h \in H$ (cf. [15], Chapter IV). Throughout all semisimple Lie algebras that we consider will be split.

The representation theory of semisimple Lie algebras of characteristic 0 has in the past received a lot of attention. All modules over semisimple Lie algebras have been classified and a great deal is known about their structure. By a theorem of Weyl all modules over a semisimple Lie algebra are completely reducible. This means that when studying modules over a semisimple Lie algebra, we can restrict our attention to the irreducible ones. Furthermore every irreducible module is uniquely determined by its highest weight.

The advent of the computer led people to implement algorithms for studying irreducible modules over semisimple Lie algebras. These algorithms were mainly concerned with computing combinatorial data concerning an irreducible module. For example Weyl's dimension formula gives an efficient algorithm for

Preprint submitted to Elsevier Preprint *7 June 2000*

computing the dimension of an irreducible module (given its highest weight), and Freudenthal's formula provides an efficient method for computing the dimensions of the weight spaces (see [4], [13], [18]).

In this paper we consider the problem of constructing an irreducible module over a semisimple Lie algebra. We suppose that we are given a semisimple Lie algebra L of characteristic 0, with a split Cartan subalgebra H and corresponding root system Φ, along with a dominant weight λ. The problem is to construct an irreducible L-module $V(\lambda)$ of highest weight λ. This means that we want a basis $\{v_1, \dots, v_n\}$ of $V(\lambda)$, together with a method for expressing $x \cdot v_i$ (for $x \in L$) as a linear combination of the basis elements v_k.

Some solutions to this problem are known. We mention the theory of Young tableaux (see, e.g., [10]), of Gelfand-Zetlin patterns ([12], [2], [3]) and Littelmann's algorithm ([17]). However, these methods typically only work for certain classes of simple Lie algebras. In this paper we describe an algorithm that works in a generic way for all semisimple Lie algebras. It roughly works as follows. Let L be a semisimple Lie algebra, and let λ be a dominant weight. We consider the left action of L on its universal enveloping algebra $U(L)$. We construct the Verma module module $A(\lambda) = U(L)/J(\lambda)$, where $J(\lambda)$ is a certain L-submodule of $U(L)$. The L-module $A(\lambda)$ is an algebra in its own right and we construct a certain left ideal $I(\lambda)$ of it, with the property that $A(\lambda)/I(\lambda)$ is an irreducible L-module of highest weight λ. To construct this last quotient we calculate a Gröbner basis of $I(\lambda)$.

This paper is organized as follows. Section 2 is devoted to the problem of calculating inside the universal enveloping algebra. For a special type of basis of the universal enveloping algebra we show how to calculate a product of two basis elements efficiently. Section 3 contains a short review of Gröbner bases for left ideals of universal enveloping algebras. Then in Section 4 we describe an algorithm for constructing an irreducible L-module of highest weight λ. We describe the form of the elements of the reduced Gröbner basis of the ideal $I(\lambda)$. This description is then used to formulate an algorithm for calculating the reduced Gröbner basis of $I(\lambda)$. In Section 5 we investigate the complexity of the algorithm. It turns out that the algorithm runs in polynomial time if the Lie algebra is fixed. Finally in Section 6 we give an example and we give an account of experiences with the implementation of the algorithm in GAP4.

Throughout this paper we make use of the standard facts on the structure of semisimple Lie algebras and their representations. These can for instance be found in [6], [14], [15].

2 Computing in the universal enveloping algebra

In this section we consider the universal enveloping algebra $U(L)$ of a semisimple Lie algebra L. We give explicit commutation formulas that allow us to compute products in $U(L)$ efficiently.

Let L be a semisimple Lie algebra of characteristic 0. Let H be a split Cartan subalgebra of L, and let Φ be the corresponding root system. Throughout $\Delta = \{\alpha_1, \dots, \alpha_l\}$ is a fixed simple system of Φ. We denote the Killing form of L by κ_L (i.e., $\kappa_L(x,y) = \mathrm{Tr}(\mathrm{ad}\, x \cdot \mathrm{ad}\, y)$). For $\alpha \in H^*$ we let $\tilde{h}_\alpha$ be the unique element of H satisfying $\alpha(h) = \kappa_L(\tilde{h}_\alpha, h)$ for all $h \in H$. The Killing form gives rise to an inner product on H^*, defined by $(\alpha, \beta) = \kappa_L(\tilde{h}_\alpha, \tilde{h}_\beta)$, for $\alpha, \beta \in H^*$. For $\alpha \in H^*$ we let $h_\alpha \in H$ denote the element

$$h_\alpha = \frac{2\tilde{h}_\alpha}{(\alpha, \alpha)}.$$

Also put $h_i = h_{\alpha_i}$ for $1 \le i \le l$. For $\alpha \in \Phi$ there are elements $x_\alpha \in L$ such that

$$\begin{aligned}
&[h_i, x_\alpha] = \alpha(h_i) x_\alpha \text{ for } 1 \le i \le l \text{ and } \alpha \in \Phi,\\
&[x_\alpha, x_{-\alpha}] = h_\alpha = \sum_{i=1}^{l} n_i^\alpha h_i \text{ where all } n_i^\alpha \in \mathbb{Z},\\
&\text{if } \alpha, \beta \in \Phi \text{ such that } \alpha \neq \pm\beta, \text{ then}\\
&\qquad [x_\alpha, x_\beta] = 0 \text{ if } \alpha + \beta \notin \Phi, \text{ and}\\
&\qquad [x_\alpha, x_\beta] = N_{\alpha,\beta} x_{\alpha+\beta} \text{ if } \alpha + \beta \in \Phi, \text{ where } N_{\alpha,\beta} \in \mathbb{Z}.
\end{aligned} \tag{1}$$

A set of elements $\{x_\alpha \mid \alpha \in \Phi\} \cup \{h_1, \dots, h_l\}$ satisfying the relations (1) is called a Chevalley basis of L. In the sequel we work with a fixed Chevalley basis of L.

We use a basis of $U(L)$ as described in [6, Ch. VIII, §12], [14, §26]. First we set for $n \ge 0$, and $\alpha \in \Phi$,

$$x_\alpha^{(n)} = \frac{x_\alpha^n}{n!},$$

and for $h \in H$ and $k \ge 0$ we set

$$\binom{h}{k} = \frac{h(h-1)\cdots(h-k+1)}{k!}.$$

Now the elements

$$\left(\prod_{\alpha>0} x_{-\alpha}^{(n_\alpha)}\right) \binom{h_1}{k_1} \cdots \binom{h_l}{k_l} \left(\prod_{\alpha>0} x_\alpha^{(m_\alpha)}\right) \tag{2}$$

form a basis of $U(L)$ (cf. [14], §26.4). We call an element of the form (2) a monomial, and we consider the problem of expressing the product of two monomials as a linear combination of monomials. A straightforward algorithm for this uses the commutation relations following from (1). However, this is quite tedious for all but the monomials of small degree. We derive explicit formulas for commuting elements of the form $x_\alpha^{(n)}$ and $\binom{h}{k}$ directly.

First we derive a formula for commuting $x_\alpha^{(m)}$ and $x_\beta^{(n)}$, where $\alpha, \beta \in \Phi$ are such that $\alpha \neq \pm\beta$.

The set of all roots that are linear combinations of α, β forms a root system Ψ of rank 2 (cf. [7], Lemma 3.6.3). So it is of type $A_1 \oplus A_1$, A_2, B_2 or G_2. By $P_{\alpha,\beta}$ we denote the set of roots of the form $i\alpha + j\beta$ where $i, j \geq 0$. By inspecting the root systems of rank 2 we see that for the set $P_{\alpha,\beta}$ there are the following possibilities:

I $P_{\alpha,\beta} = \{\alpha, \beta\}$,
II $P_{\alpha,\beta} = \{\alpha, \beta, \alpha+\beta\}$,
III $P_{\alpha,\beta} = \{\alpha, \beta, \alpha+\beta, \alpha+2\beta\}$,
IV $P_{\alpha,\beta} = \{\alpha, \beta, \alpha+\beta, 2\alpha+\beta\}$,
V $P_{\alpha,\beta} = \{\alpha, \beta, \alpha+\beta, 2\alpha+\beta, 3\alpha+\beta, 3\alpha+2\beta\}$,
VI $P_{\alpha,\beta} = \{\alpha, \beta, \alpha+\beta, \alpha+2\beta, \alpha+3\beta, 2\alpha+3\beta\}$,
VII $P_{\alpha,\beta} = \{\alpha, \beta, \alpha+\beta, 2\alpha+\beta, \alpha+2\beta\}$.

The sets $P_{\alpha,\beta}$ of types V, VI and VII only occur when Ψ is of type G_2. The types III and IV can occur when Ψ is of type B_2 or G_2. Type II can occur when Ψ is of type A_2, B_2 or G_2, and type I can always occur regardless of the type of Ψ.

Lemma 1 *The product $x_\beta^{(m)} x_\alpha^{(n)}$ is given by the following formulas:*

$$\text{I } x_\beta^{(m)} x_\alpha^{(n)} = x_\alpha^{(n)} x_\beta^{(m)},$$

$$\text{II } x_\beta^{(m)} x_\alpha^{(n)} = \sum_{k=0}^{\min(m,n)} (-1)^k N_{\alpha,\beta}^k x_\alpha^{(n-k)} x_\beta^{(m-k)} x_{\alpha+\beta}^{(k)},$$

$$\text{III } x_\beta^{(m)} x_\alpha^{(n)} = \sum_{\substack{k,l\geq 0\\ n-k-l\geq 0\\ m-k-2l\geq 0}} (-1)^k c_{\alpha,\beta}^{k+l} c_{\beta,\alpha+\beta}^l x_\alpha^{(n-k-l)} x_\beta^{(m-k-2l)} x_{\alpha+\beta}^{(k)} x_{\alpha+2\beta}^{(l)},$$

where $c_{\alpha,\beta} = N_{\alpha,\beta}$ *and* $c_{\beta,\alpha+\beta} = \dfrac{1}{2} N_{\beta,\alpha+\beta}$,

$$\text{IV } x_\beta^{(m)} x_\alpha^{(n)} = \sum_{\substack{k,l\geq 0\\ n-k-2l\geq 0\\ m-k-l\geq 0}} (-1)^k c_{\alpha,\beta}^{k+l} c_{\alpha,\alpha+\beta}^{l} x_\alpha^{(n-k-2l)} x_\beta^{(m-k-l)} x_{\alpha+\beta}^{(k)} x_{2\alpha+\beta}^{(l)}$$

where $c_{\alpha,\beta} = N_{\alpha,\beta}$ *and* $c_{\alpha,\alpha+\beta} = \frac{1}{2} N_{\alpha,\alpha+\beta}$,

$$\text{V } x_\beta^{(m)} x_\alpha^{(n)} = \sum_{\substack{p,q,r,s\geq 0\\ n-p-2q-3r-3s\geq 0\\ m-p-q-r-2s\geq 0}} (-1)^{p+r} c_{\alpha,\beta}^{p+q+r+s} c_{\alpha,\alpha+\beta}^{q+r+s} c_{\alpha,2\alpha+\beta}^{r} c_0^s \cdot$$
$$x_\alpha^{(n-p-2q-3r-3s)} x_\beta^{(m-p-q-r-2s)} x_{\alpha+\beta}^{(p)} x_{2\alpha+\beta}^{(q)} x_{3\alpha+\beta}^{(r)} x_{3\alpha+2\beta}^{(s)},$$

where $c_{\alpha,\beta} = N_{\alpha,\beta}$, $c_{\alpha,\alpha+\beta} = \frac{1}{2} N_{\alpha,\alpha+\beta}$, $c_{\alpha,2\alpha+\beta} = \frac{1}{3} N_{\alpha,2\alpha+\beta}$,

and $c_0 = \frac{1}{2}(N_{\alpha,\beta} N_{\alpha+\beta,2\alpha+\beta} + c_{\alpha,2\alpha+\beta} N_{\beta,3\alpha+\beta})$,

$$\text{VI } x_\beta^{(m)} x_\alpha^{(n)} = \sum_{\substack{p,q,r,s\geq 0\\ n-p-q-r-2s\geq 0\\ m-p-2q-3r-3s\geq 0}} (-1)^{p+r} 2^s c_{\alpha,\beta}^{p+q+r+2s} c_{\beta,\alpha+\beta}^{q+r+s} c_{\alpha+\beta,\alpha+2\beta}^{s} c_{\beta,\alpha+2\beta}^{r} \cdot$$
$$x_\alpha^{(n-p-q-r-2s)} x_\beta^{(m-p-2q-3r-3s)} x_{\alpha+\beta}^{(p)} x_{\alpha+2\beta}^{(q)} x_{\alpha+3\beta}^{(r)} x_{2\alpha+3\beta}^{(s)},$$

where $c_{\alpha,\beta} = N_{\alpha,\beta}$, $c_{\beta,\alpha+\beta} = \frac{1}{2} N_{\beta,\alpha+\beta}$, $c_{\beta,\alpha+2\beta} = \frac{1}{3} N_{\beta,\alpha+2\beta}$,

and $c_{\alpha+\beta,\alpha+2\beta} = \frac{1}{3} N_{\alpha+\beta,\alpha+2\beta}$,

$$\text{VII } x_\beta^{(m)} x_\alpha^{(n)} = \sum_{\substack{p,q,r\geq 0\\ n-p-2q-r\geq 0\\ m-p-q-2r\geq 0}} (-1)^{p} c_{\alpha,\beta}^{p+q+r} c_{\alpha,\alpha+\beta}^{(q)} c_{\beta,\alpha+\beta}^{(r)}$$
$$x_\alpha^{(n-p-2q-r)} x_\beta^{(m-p-q-2r)} x_{\alpha+\beta}^{(p)} x_{2\alpha+\beta}^{(q)} x_{\alpha+2\beta}^{(r)}$$

where $c_{\alpha,\beta} = N_{\alpha,\beta}$, $c_{\alpha,\alpha+\beta} = \frac{1}{2} N_{\alpha,\alpha+\beta}$, $c_{\beta,\alpha+\beta} = \frac{1}{2} N_{\beta,\alpha+\beta}$.

PROOF. First of all I is trivial. The proofs of the other formulas are by induction. By induction on n we have

$$x_\beta x_\alpha^{(n)} = \sum_{k=0}^{n} \frac{(-1)^k}{k!} \left(\prod_{i=1}^{k} N_{\alpha,(i-1)\alpha+\beta} \right) x_\alpha^{(n-k)} x_{k\alpha+\beta},$$

which holds for all types. The formulas for $m = 1$ follow immediately from this. Now in order to perform the induction step it is convenient to define $x_\gamma^{(k)} = 0$ if $k < 0$, for all roots γ, and let the indices in the summations run through $\mathbb{Z}$. The induction step is similar for all formulas. We prove IV. In this case the formula for $m = 1$ reads

$$x_\beta x_\alpha^{(n)} = x_\alpha^{(n)} x_\beta - c_{\alpha,\beta} x_\alpha^{(n-1)} x_{\alpha+\beta} + c_{\alpha,\beta} c_{\alpha,\alpha+\beta} x_\alpha^{(n-2)} x_{2\alpha+\beta}.$$

Furthermore, by induction,

$$\begin{aligned} x_\beta^{(m+1)} x_\alpha^{(n)} &= \frac{1}{m+1} x_\beta x_\beta^{(m)} x_\alpha^{(n)} \\ &= \frac{1}{m+1} \sum_{k,l\in\mathbb{Z}} (-1)^k c_{\alpha,\beta}^{k+l} c_{\alpha,\alpha+\beta}^{l} x_\beta x_\alpha^{(n-k-2l)} x_\beta^{(m-k-l)} x_{\alpha+\beta}^{(k)} x_{2\alpha+\beta}^{(l)}. \end{aligned}$$

Now by using the formula for $m=1$ we see that

$$\begin{aligned} & x_\beta x_\alpha^{(n-k-2l)} x_\beta^{(m-k-l)} x_{\alpha+\beta}^{(k)} x_{2\alpha+\beta}^{(l)} \\ &\quad = \left(x_\alpha^{(n-k-2l)} x_\beta - c_{\alpha,\beta} x_\alpha^{(n-k-2l-1)} x_{\alpha+\beta} + c_{\alpha,\beta} c_{\alpha,\alpha+\beta} x_\alpha^{(n-k-2l-2)} x_{2\alpha+\beta} \right) \\ &\qquad\qquad \times x_\beta^{(m-k-l)} x_{\alpha+\beta}^{(k)} x_{2\alpha+\beta}^{(l)}. \end{aligned}$$

And (as $x_\beta x_\beta^{(r)} = (r+1) x_\beta^{(r+1)}$) this is equal to

$$(m+1-k-l) x_\alpha^{(n-k-2l)} x_\beta^{(m+1-k-l)} x_{\alpha+\beta}^{(k)} x_{2\alpha+\beta}^{(l)} \tag{3}$$
$$- c_{\alpha,\beta}(k+1) x_\alpha^{(n-k-2l-1)} x_\beta^{(m-k-l)} x_{\alpha+\beta}^{(k+1)} x_{2\alpha+\beta}^{(l)} \tag{4}$$
$$+ c_{\alpha,\beta} c_{\alpha,\alpha+\beta} (l+1) x_\alpha^{(n-k-2l-2)} x_\beta^{(m-k-l)} x_{\alpha+\beta}^{(k)} x_{2\alpha+\beta}^{(l+1)}. \tag{5}$$

So $x_\beta^{(m+1)} x_\alpha^{(n)}$ is equal to the sum of three summations, where k,l run over $\mathbb{Z}$. Now we substitute k for $k+1$ in the summation arising from (4) and l for $l+1$ in the summation arising from (5). Then we add, and obtain only one summation, and the factor $m+1$ cancels. We note that if k,l are such that $x_\alpha^{(n-k-2l)} x_\beta^{(m-k-l)} x_{\alpha+\beta}^{(k)} x_{2\alpha+\beta}^{(l)} = 0$, then (3), (4) and (5) are zero as well. $\square$

Now we deal with commuting $x_\alpha^{(m)}$ and $x_{-\alpha}^{(n)}$, and $x_\alpha^{(m)}$ and $\binom{h}{k}$. The following lemma is [14], Lemma 26.2.

Lemma 2 *We have the formula*

$$x_\alpha^{(m)} x_{-\alpha}^{(n)} = \sum_{k=0}^{\min(m,n)} x_{-\alpha}^{(n-k)} \binom{h_\alpha - n - m + 2k}{k} x_\alpha^{(m-k)}.$$

Lemma 3 *For $h \in H$ and $r \in \mathbb{Z}$ we have*

$$\binom{h+r}{m} x_{-\alpha}^{(n)} = x_{-\alpha}^{(n)} \binom{h+r-n\alpha(h)}{m},$$

and

$$x_\alpha^{(n)} \binom{h+r}{m} = \binom{h+r-n\alpha(h)}{m} x_\alpha^{(n)}.$$

PROOF. Let $f \in F[X]$ be a univariate polynomial. Then by Lemma 26.3.D of [14] we have that $x_\alpha^{(n)} f(h) = f(h - n\alpha(h)) x_\alpha^{(n)}$. The second formula is an immediate consequence of this. The first formula follows from $f(h)x_{-\alpha}^{(n)} = x_{-\alpha}^{(n)} f(h - n\alpha(h))$. $\square$

The following lemma is immediate.

Lemma 4

$$x_\alpha^{(n)} x_\alpha^{(m)} = \binom{n+m}{m} x_\alpha^{(n+m)}.$$

By applying the formulas of Lemma 2 and 3, we get factors of the form

$$\binom{\sum_{i=1}^{l} n_i h_i + r}{k}$$

where the n_i, r and k are integers. We can rewrite such factors by applying the well-known formula

$$\binom{a+b}{k} = \sum_{i=0}^{k} \binom{a}{i} \binom{b}{k-i}. \tag{6}$$

Furthermore, for rewriting products of the form $\binom{h_i}{l}\binom{h_i}{k}$ we use the following formula, which is straightforward to prove by induction:

$$\binom{h}{l}\binom{h}{k} = \sum_{i=0}^{l} \binom{l}{i} \binom{k+i}{l} \binom{h}{k+i} \quad \text{where } l \leq k. \tag{7}$$

Let a, a' be two monomials of the form (2) with exponents n_α, k_i, m_α and $n'_\alpha, k'_i, m'_\alpha$ respectively. By using the formulas of Lemmas 1, 2, 3, and 4 together with (6) and (7), we express the product $a \cdot a'$ as a linear combination of monomials of the form (2). This gives us a general formula for calculating the product of any two monomials in $U(L)$. We call this formula the *multiplication law* of $U(L)$.

Example 5 *Let L be the Lie algebra of type A_2. Let $\alpha_1, \alpha_2, \alpha_3 = \alpha_1 + \alpha_2$ be the positive roots, and write $y_i = x_{-\alpha_i}$ for $i = 1, 2, 3$. Let N^- be the subalgebra of L generated by the y_i. The universal enveloping algebra $U(N^-)$ will be of paramount importance in the sequel. In this case we can choose the structure constants of L so that $[y_1, y_2] = y_3$. We get the multiplication law of $U(N^-)$ by a single application of formula II of Lemma 1, and some applications of*

Lemma 4; it reads

$$y_1^{(n_1)}y_2^{(n_2)}y_3^{(n_3)}y_1^{(m_1)}y_2^{(m_2)}y_3^{(m_3)} = \sum_{\substack{k\geq 0\\ m_1-k\geq 0\\ n_2-k\geq 0}} (-1)^k \binom{n_1+m_1-k}{n_1}\binom{n_2+m_2-k}{m_2}\binom{n_3+m_3}{m_3}\binom{n_3+m_3+k}{n_3+m_3} \times y_1^{(n_1+m_1-k)}y_2^{(n_2+m_2-k)}y_3^{(n_3+m_3+k)}.$$

3 Gröbner bases for left ideals in $U(N^-)$

In this section we let $\alpha_1,\dots,\alpha_s$ denote the positive roots of Φ, where, as before, $\alpha_1,\dots,\alpha_l$ are the simple roots. Furthermore, to ease notation a little, we set $y_i = x_{-\alpha_i}$. We briefly review an approach to Gröbner bases for left ideals in $U(N^-)$, where N^- is the subalgebra of L spanned by the y_i. For this we follow ideas presented in [1], [16]. For more details and proofs we refer to these papers.

Throughout we use the basis $\{y_1^{(n_1)}\cdots y_s^{(n_s)}\}$ of $U(N^-)$ and call an element $y_1^{(n_1)}\cdots y_s^{(n_s)}$ a *monomial*. The degree of such a monomial is the number $n_1+\cdots+n_s$. We fix an ordering $<$ of the monomials. Relative to this ordering every element $f\in U(N^-)$ has a leading monomial, which is the biggest monomial occurring in f. It is denoted by $\mathrm{LM}(f)$. We assume that $<$ satisfies the descending chain condition (i.e., there are no infinite descending chains $a_1 < a_2 < a_3\cdots$ of monomials), and that $<$ is multiplicative (i.e., if $a<b$, then $\mathrm{LM}(ca) < \mathrm{LM}(cb)$ for all monomials c). Also we assume that $<$ is compatible with the degree, i.e., $\deg(a) < \deg(b)$ implies that $a<b$. An ordering $<$ that is multiplicative, satisfies the descending chain condition and is degree compatible is called a *term ordering*. An example of a term ordering is the deglex ordering $<_{\mathrm{dlex}}$. If $\deg(a) < \deg(b)$, then $a<_{\mathrm{dlex}}b$. If $\deg(a)=\deg(b)$, then we write

$$a = y_1^{(n_1)}\cdots y_s^{(n_s)} \text{ and } b = y_1^{(m_1)}\cdots y_s^{(m_s)}. \tag{8}$$

In this case $a<_{\mathrm{dlex}}b$ if the first nonzero integer in the list (n_i-m_i) is positive.

Let a,b be two monomials; then we say that a is a factor of b (or b is divisible by a), if there is a monomial c such that $\mathrm{LM}(ca)=b$. Again write a,b as in (8). Since $<$ is degree compatible we have that a is a factor of b if and only if $n_i\leq m_i$ for $1\leq i\leq s$. In this case $c = y_1^{(m_1-n_1)}\cdots y_s^{(m_s-n_s)}$ satisfies $\mathrm{LM}(ca)=b$.

Let $G\subset U(N^-)$ and $f\in U(N^-)$ and suppose that there is an element $g\in G$

such that $\mathrm{LM}(g)$ divides a monomial a occurring in f. Let c be a monomial such that $\mathrm{LM}(cg) = a$. Let γ be the coefficient of a in f and δ the coefficient of $\mathrm{LM}(cg)$ in cg. Then we say that f reduces modulo G to $h = f - \frac{\gamma}{\delta}cg$. As $<$ satisfies the descending chain condition, any sequence of reduction steps terminates with an element that cannot be reduced further modulo G. We call this element a normal form of f modulo G.

Let I be the left ideal of $U(N^-)$ generated by the set G. Then we say that G is a Gröbner basis of I if for all $f \in I$ there is a $g \in G$ such that $\mathrm{LM}(g)$ divides $\mathrm{LM}(f)$. It is straightforward to see that reduction of an element $f \in U(N^-)$ modulo a Gröbner basis of I always terminates with a unique result. We call this result the normal form of f modulo I.

Let $f, g \in U(N^-)$ and write

$$\mathrm{LM}(f) = y_1^{(n_1)} \cdots y_s^{(n_s)} \text{ and } \mathrm{LM}(g) = y_1^{(m_1)} \cdots y_s^{(m_s)}.$$

For an integer k we define $[k] = k$ if $k \geq 0$, and $[k] = 0$ if $k \leq 0$. Now set $a = y_1^{([n_1-m_1])} \cdots y_s^{([n_s-m_s])}$, and $b = y_1^{([m_1-n_1])} \cdots y_s^{([m_s-n_s])}$. Let γ, δ be the coefficients of the leading monomials of bf and ag respectively. Then

$$S(f, g) = \delta bf - \gamma ag$$

is called the S-element of f and g. For a proof of the next theorem we refer to [1], [13], [16].

Theorem 6 *Let I be a left ideal of $U(N^-)$ generated by the set G. Then G is a Gröbner basis of I if and only if $S(g_1, g_2)$ reduces to zero modulo G for all $g_1, g_2 \in G$.*

Theorem 6 leads to the usual algorithm for calculating a Gröbner basis. We start with a set G generating a left ideal of $U(N^-)$. If all S-elements $S(g_1, g_2)$ for $g_1, g_2 \in G$ reduce to zero modulo G, then G is a Gröbner basis. If not, then we let h be a normal form modulo G of $S(g_1, g_2)$, where $g_1, g_2 \in G$ are such that this normal form is non-zero. We add h to G and repeat the process. In [1], [13], [16] it is shown that this process terminates, at which point the set G is necessarily a Gröbner basis.

Let G be a Gröbner basis of a left ideal of $U(L)$. Then G is called *reduced* if

(1) the coefficient of $\mathrm{LM}(g)$ in g is 1 for all $g \in G$,
(2) for all $g_1, g_2 \in G$ such that $g_1 \neq g_2$ we have that $\mathrm{LM}(g_2)$ does not divide any monomial in g_1.

Lemma 7 *Let I be a finitely generated left ideal of $U(N^-)$. With respect to a fixed term ordering I has a unique reduced Gröbner basis.*

PROOF. The proof of this result is exactly the same as the proof for the analogous result for ideals in a polynomial ring (see, e.g., [8], §7.2). Existence is proved by starting with an arbitrary finite Gröbner basis $\tilde{G}$, and replacing $g \in \tilde{G}$ by a normal form of g modulo $\tilde{G} \setminus \{g\}$, until the second condition for reducedness holds. Then the first condition is enforced by dividing all elements by a suitable scalar. $\square$

4 Constructing a highest-weight module

Let L be a semisimple Lie algebra of characteristic 0, with root system Φ and Chevalley basis $\{x_\alpha \mid \alpha \in \Phi\} \cup \{h_1, \ldots, h_l\}$. By P we denote the weight lattice of Φ. In this section we describe an algorithm for constructing the irreducible highest-weight module over L with highest weight λ, where $\lambda \in P$ is a dominant weight.

We start off with the universal enveloping algebra $U(L)$ and let L act on the left by left multiplication. Let $J(\lambda)$ be the left ideal of $U(L)$ generated by the elements x_α for $\alpha > 0$ together with $h_i - \lambda(h_i)$ for $1 \leq i \leq l$. We consider the quotient $A(\lambda) = U(L)/J(\lambda)$ (which is called a Verma module). It is straightforward to see that the generators of $J(\lambda)$ in fact form a Gröbner basis of $J(\lambda)$ (using Theorem 6, which holds for the universal enveloping algebra of any Lie algebra, not just for $U(N^-)$). Therefore the Verma module $A(\lambda)$ is spanned by the cosets of the monomials of the form (2) that are not divisible by any leading monomial of an element of this Gröbner basis. These are precisely the monomials $y_1^{(n_1)} \cdots y_s^{(n_s)}$ (notation as in Section 3). This means that there is a natural isomorphism of vector spaces $\phi : A(\lambda) \to U(N^-)$. We let N^- act on $U(N^-)$ by left multiplication. Then ϕ is an isomorphism of N^--modules. Now since $A(\lambda)$ is an L-module, we can give $U(N^-)$ an L-module structure by $z \cdot a = \phi(z \cdot \phi^{-1}(a))$ for $z \in L$ and $a \in U(N^-)$. Note that this is compatible with the natural left action of N^- on $U(N^-)$. Also we can use ϕ to make $A(\lambda)$ into an associative algebra. Then $A(\lambda)$ and $U(N^-)$ are isomorphic as associative algebras, and as L-modules. In the sequel we will use both $U(N^-)$ and $A(\lambda)$ to denote the same module.

Let $I(\lambda)$ be the left ideal of $A(\lambda)$ generated by the elements $y_i^{(\lambda(h_i)+1)}$ for $1 \leq i \leq l$. Furthermore we let $\overline{I(\lambda)}$ be the L-submodule of $A(\lambda)$ generated by $I(\lambda)$. The next result describes the object where we are after.

Lemma 8 *$A(\lambda)/\overline{I(\lambda)}$ is a finite-dimensional irreducible L-module with highest weight λ.*

PROOF. This can be proved in exactly the same way as [14], Theorem

21.4. □

Proposition 9 *We have* $I(\lambda) = \overline{I(\lambda)}$.

PROOF. Set $W = \overline{I(\lambda)}$; then W is an infinite-dimensional L-module. Set $w_i = y_i^{(\lambda(h_i)+1)}$ for $1 \leq 1 \leq l$. Then the w_i are weight vectors of weights $\lambda - (\lambda(h_i)+1)\alpha_i$. Set $x_j = x_{\alpha_j}$ for $1 \leq j \leq l$. We claim that $x_j \cdot w_i = 0$ for $1 \leq j \leq l$. First if $j \neq i$, then $[x_j, y_i] = 0$ so that $x_j \cdot w_i = y_i^{(\lambda(h_i)+1)} x_j = 0$ (in $A(\lambda)$). Secondly, by induction it is straightforward to show that

$$x_i y_i^{(k+1)} = y_i^{(k+1)} x_i + y_i^{(k)}(h_i - k)$$

(cf. [14], Lemma 21.2). This implies that $x_i y_i^{(\lambda(h_i)+1)} \in J(\lambda)$. So w_i is a highest-weight vector, and therefore the L-submodule W_i of $A(\lambda)$ generated by w_i is spanned by the weight vectors $y_1^{(n_1)} \cdots y_s^{(n_s)} \cdot w_i$. In particular, $W_i \subset I(\lambda)$ (note that here we use the fact that the left action of N^- on $A(\lambda)$ coincides with the left multiplication). Hence $W_1 + \cdots + W_l$ is an L-submodule of $A(\lambda)$ and it contains w_i for $1 \leq i \leq l$. Therefore it must be equal to W, and hence $W = I(\lambda)$. □

We set $V(\lambda) = A(\lambda)/I(\lambda)$; then by Lemma 8, and Proposition 9, $V(\lambda)$ is the irreducible highest-weight module over L with highest weight λ. The algorithm for computing a basis of $V(\lambda)$ first computes a Gröbner basis G of $I(\lambda)$. Then the monomials in $A(\lambda)$ that are not divisible by $\mathrm{LM}(g)$ for $g \in G$, form a basis of $V(\lambda)$. We show that $I(\lambda)$ has a Gröbner basis of a special form, that allows us to compute it rather efficiently. In the sequel we use a fixed term ordering $<$ on the monomials of $A(\lambda)$.

We recall that P denotes the weight lattice of Φ. A weight $\mu \in P$ is said to be a *weight of* $V(\lambda)$ if the weight space of μ in $V(\lambda)$ is non-zero. We note that there are efficient algorithms for calculating the weights of $V(\lambda)$ and their multiplicities (i.e., the dimensions of the corresponding weight spaces), see, e.g., [4], [13], [18].

Now we define the *extended weight diagram* of λ to be the smallest subset $D \subset P$ such that

(1) D contains the weights of $V(\lambda)$ and
(2) if μ is a weight of $V(\lambda)$ and $\alpha > 0$ is a positive root, then $\mu - \alpha \in D$.

All weights $\mu \in D$ can be written as $\mu = \lambda - \sum_{i=1}^{l} k_i \alpha_i$, where the k_i are non-negative integers (note that the summation is only over the simple roots). The *level* of μ is the number $\sum_{i=1}^{l} k_i$. So λ is the unique weight of level 0.

If $v \in V(\lambda)$ is a weight vector of weight μ, then $y_i \cdot v$ is a weight vector of weight $\mu - \alpha_i$ (this follows from an elementary calculation). Therefore we define the *weight* of a monomial $y_1^{(n_1)} \cdots y_s^{(n_s)}$ to be

$$\lambda - \sum_{i=1}^{s} n_i \alpha_i.$$

An element $f \in A(\lambda)$ is called *homogeneous* if all its monomials are of the same weight μ. In this case f is said to be of weight μ.

Lemma 10 *The ideal $I(\lambda)$ is spanned by homogeneous elements.*

PROOF. Let W_i be the space of the proof of Proposition 9. This space is spanned by the elements $y_1^{(n_1)} \cdots y_s^{(n_s)} \cdot w_i$, and these are homogeneous. So as $I(\lambda) = W_1 + \cdots + W_l$ we have that $I(\lambda)$ is spanned by homogeneous elements. $\Box$

Let R be a basis of $I(\lambda)$ consisting of homogeneous elements. For a weight $\mu \in P$ we write R_μ for the set of elements of R that are of weight μ. Note that R_μ is a finite set. By (maybe) taking linear combinations of the elements of R_μ we get a set $\tilde{R}_\mu$ spanning the same space as R_μ and such that $\mathrm{LM}(f_1) \neq \mathrm{LM}(f_2)$ for $f_1, f_2 \in \tilde{R}_\mu$ (i.e., $\tilde{R}_\mu$ is a set in echelon form).

Lemma 11 *Let D be the extended weight diagram of λ. Then $\tilde{G} = \bigcup_{\mu \in D} \tilde{R}_\mu$ is a Gröbner basis of $I(\lambda)$.*

PROOF. Let $f \in I(\lambda)$. We must prove that $\mathrm{LM}(f)$ is divisible by a $\mathrm{LM}(g)$ for a $g \in \tilde{G}$. Write $f = f_1 + \cdots + f_k$, where the f_i are homogeneous, and $f_i \in I(\lambda)$. The leading monomial of f occurs in precisely one of the f_i, say in f_1. Let μ be the weight of f_1. If $\mu \in D$, then f_1 lies in the space spanned by $\tilde{R}_\mu$. So because $\tilde{R}_\mu$ is in echelon form, there is a $g \in \tilde{R}_\mu$ such that $\mathrm{LM}(g) = \mathrm{LM}(f_1)$. If $\mu \notin D$, then μ is not a weight of $V(\lambda)$ and hence $\mathrm{LM}(f_1)$ lies in $I(\lambda)$. Write $\mathrm{LM}(f_1) = y_1^{(n_1)} \cdots y_s^{(n_s)}$. Let j be an index such that $n_j > 0$. As $\mu \notin D$ we have that $\mu + \alpha_j$ is not a weight of $V(\lambda)$. Therefore the element

$$a = y_1^{(n_1)} \cdots y_{j-1}^{(n_{j-1})} y_j^{(n_j - 1)} y_{j+1}^{(n_{j+1})} \cdots y_s^{(n_s)}$$

lies in $I(\lambda)$. Note that the level of $\mu + \alpha_j$ is strictly smaller than the level of μ. So by induction on the level, we have that there is a $g \in \tilde{G}$ such that $\mathrm{LM}(g)$ is a factor of a. Hence $\mathrm{LM}(g)$ is also a factor of $\mathrm{LM}(f_1) = \mathrm{LM}(f)$. $\Box$

Corollary 12 *Let D be the extended weight diagram of λ. Then the reduced Gröbner basis G of $I(\lambda)$ relative to $<$ consists of homogeneous elements and the weight of each element lies in D.*

PROOF. Let $\tilde{G}$ be a Gröbner basis of $I(\lambda)$ as in Lemma 10. Then $\tilde{G}$ consists of homogeneous elements and the weight of each element lies in D. From $\tilde{G}$ we construct a reduced Gröbner basis G following the procedure outlined in the proof of Lemma 7. Let $g \in \tilde{G}$ and let h be a normal form of g modulo $\tilde{G} \setminus \{g\}$. Then h is homogeneous and either h is zero, or h is of the same weight as g. Hence the statement of the corollary follows. $\Box$

The following lemma is handy.

Lemma 13 *Let $a = y_1^{(n_1)} \cdots y_s^{(n_s)}$ and $b = y_1^{(m_1)} \cdots y_s^{(m_s)}$ be two monomials of $U(N^-)$ such that $a \neq b$. If a divides b, then the weight of b is of strictly higher level than the weight of a.*

PROOF. Since a divides b we have that $n_i \leq m_i$ for $1 \leq i \leq s$. As $a \neq b$ at least one inequality is strict. $\Box$

Let D, G be as in Corollary 12. For $\mu \in D$ we let M_μ be the set of monomials a of weight μ such that a is not divisible by the leading monomial of any element $g \in G$ such that the weight of g is of lower level than μ. Let μ be a weight of $V(\lambda)$. Then by m_μ we denote the multiplicity of μ (i.e., the dimension of the weight space of μ in $V(\lambda)$).

Proposition 14 *Let D, G be as in Corollary 12. For $\mu \in D$ let G_μ denote the set of all elements of G that are of weight μ. If μ is not a weight of $V(\lambda)$, then $G_\mu = M_\mu$. If μ is a weight of $V(\lambda)$, then write $M_\mu = \{a_1, \ldots, a_{r_\mu}\}$, where $a_1 < a_2 < \cdots < a_{r_\mu}$. Then G_μ consists of $r_\mu - m_\mu$ elements of the form*

$$a_j + \sum_{i=1}^{m_\mu} q_{ij} a_i$$

for $m_\mu + 1 \leq j \leq r_\mu$, and certain scalars q_{ij}.

PROOF. Let μ be an element of D that is not a weight of $V(\lambda)$. Then it is clear that G_μ consists of monomials of weight μ. Hence $G_\mu \subset M_\mu$ as G is reduced. For the other inclusion let $a \in M_\mu$. Then $a \in I(\lambda)$ so that a reduces to 0 modulo G. Hence there is a $g \in G$ such that $\mathrm{LM}(g)$ divides a. By the definition of M_μ we see that the level of the weight of this g is at least the level of μ. By Lemma 13 we see that this implies that the weight of g is μ. Hence g must be a itself, i.e., we have that $a \in G_\mu$.

Now suppose that μ is a weight of $V(\lambda)$. Of all monomials in M_μ exactly m_μ do not reduce modulo G, and the others do. So because no monomial in M_μ

reduces modulo a $g \in G$ that is not of weight μ, there are $r_\mu - m_\mu$ elements in G_μ. Furthermore, every monomial of M_μ that does reduce modulo G occurs as the leading monomial of an element of G_μ. Let $b_1, \dots, b_{m_\mu}$ be the monomials of M_μ that do not reduce modulo G. Then since G is reduced a $g \in G_\mu$ is of the form

$$g = \mathrm{LM}(g) + \sum_{i=1}^{m_\mu} u_i b_i$$

where the u_i are certain scalars. Therefore the b_i are the m_μ smallest elements of M_μ and the proposition follows. $\Box$

Now we formulate the algorithm for calculating the reduced Gröbner basis of $I(\lambda)$, relative to the fixed term ordering $<$. The idea of the algorithm is straightforward: we loop through the extended weight diagram D, and for each weight $\mu \in D$ we add elements of weight μ to the Gröbner basis. Since we know the dimension of the weight space of μ (e.g. from Freudenthal's formula), we also know how many elements of weight μ we have to add to the Gröbner basis. For $i \geq 0$ we let D_i be the set of all μ in D of level i. Also for a monomial $a = y_1^{(n_1)} \cdots y_s^{(n_s)}$ we set

$$y_j * a = y_1^{(n_1)} \cdots y_j^{(n_j+1)} \cdots y_s^{(n_s)}.$$

Algorithm HighestWeightModule

Input: a semisimple Lie algebra L with root system Φ, and a dominant weight λ.

Output: the reduced Gröbner basis G of $I(\lambda)$ and a set of monomials $B \subset A(\lambda)$ such that the cosets of the elements of B form a basis of $A(\lambda)/I(\lambda)$.

Step 1 (Initialization.) Compute the extended weight diagram D of λ, and set $G := \emptyset$, $B := \{1\}$. Let max-lev be the maximum level of a weight in D.

Step 2 (Loop through D.) For $1 \leq i \leq$max-lev and $\mu \in D_i$ do the following:

Step 2a (Create a set of candidate monomials.) For all $\alpha_j > 0$ such that $\mu + \alpha_j \in D$ let $M^\mu_{\alpha_j}$ be the set of all $y_j * a$, where a runs through the elements of B of weight $\mu + \alpha_j$. Set $M_\mu = \cup M^\mu_{\alpha_j}$, and erase all elements a from M_μ such that a is divisible by a leading monomial of an element of G.

Step 2b If μ is not a weight of $V(\lambda)$, then set $G = G \cup M_\mu$. Otherwise execute Steps 2c and 2d.

Step 2c (Add elements to G, and erase their leading monomials from M_μ until we have the right number of basis elements in M_μ left.) Let m_μ be the multiplicity of μ. While the size of M_μ is strictly larger than m_μ, loop through $G \times G$ and for each $(g_1, g_2) \in G \times G$ do the following:

Step 2c1 Set $s := S(g_1, g_2)$ and calculate a normal form h of s modulo G.

Step 2c2 If $h \neq 0$, then erase the leading monomial of h from M_μ, divide h by the coefficient of $\mathrm{LM}(h)$ in h and add h to G. If necessary perform further reductions to ensure that for all $g_1 \neq g_2 \in G$ we have that $\mathrm{LM}(g_1)$ does not divide any monomial in g_2.

Step 2d Set $B := B \cup M_\mu$.

Step 3 Return G, B.

Remark 15 *We note that during the algorithm G is always a set of homogeneous elements. This follows from the fact that the S-element of homogeneous elements is homogeneous again. Furthermore, by inspecting g_1, g_2 it is straightforward to predict what the weight of the S-element $S(g_1, g_2)$ will be. This enables us to construct and reduce only S-elements of the correct weight.*

Proposition 16 *On input L, λ the algorithm* HighestWeightModule *terminates. Denote the Gröbner basis in the output by G and the set of monomials by B. Then G is the reduced Gröbner basis of $I(\lambda)$ (with respect to the fixed term ordering $<$) and the cosets of the elements of B form a basis of $V(\lambda)$.*

PROOF. Let G_0 be the reduced Gröbner basis of $I(\lambda)$ (relative to the fixed term ordering $<$). By Corollary 12 the elements of G_0 are homogeneous. We denote the set of elements of G_0 that are of weight μ by $G_{0\mu}$. Similarly G_μ is the set of elements of G that are of weight μ. We show that for all $\mu \in D$ we have that $G_{0\mu} = G_\mu$. For that we use induction on the level of μ. We note that by Lemma 13 we have that after Step 2 has been executed for the weight μ, the set G_μ will not be changed. So it is enough to show that after Step 2 we have that $G_{0\mu} = G_\mu$.

If the level of μ is 0 (i.e., $\mu = \lambda$), then the statement is trivial as there are no elements of weight μ in G_0, nor in G. So suppose that $G_\nu = G_{0\nu}$ for all $\nu \in D$ that are of strictly smaller level than μ. If μ is not a weight of $V(\lambda)$ then we add to G the set of all monomials M_μ that are of weight μ and are not divisible by a leading monomial of any g that is already contained in G. Now by the induction hypothesis and Proposition 14 we have that $M_\mu = G_{0\mu}$.

Now suppose that μ is a weight of $V(\lambda)$. Then we construct S-elements to find the elements of G. Note that there are no pairs $(g_1, g_2) \in G \times G$ such that $\mathrm{LM}(g_1)$ divides $\mathrm{LM}(g_2)$. Hence the weight of the S-element $S(g_1, g_2)$ is of strictly higher level than the maximum of the levels of the weights of g_1, g_2. Or, equivalently, all S-elements of weight μ are of the form $S(g_1, g_2)$, where the weights of g_1, g_2 are of smaller level than the level of μ. Therefore all S-elements of weight μ that can be constructed using the elements of G_0 can also be constructed using the elements of G. So since the ordinary algorithm for computing a Gröbner basis (cf. Section 3) terminates, we find enough S-elements in Step 2c1 of the algorithm to construct all elements of $G_{0\mu}$.

This means that we find enough elements to reduce all but m_μ elements of M_μ. Hence Step 2c terminates. As G is being kept reduced in Step 2c2, the elements of G_μ are all of the form $a_j + \sum_{i=1}^{m_\mu} p_{ij} a_i$ (notation as in Proposition 14). It is clear that we cannot have two different elements of this form with the same a_j as leading monomial in $I(\lambda)$ (otherwise the dimension of the weight space of μ is not m_μ). Hence after Step 2c has terminated we have $G_\mu = G_{0\mu}$.

The elements of B are exactly those monomials that are not divisible by a $\mathrm{LM}(g)$ for $g \in G$. Hence the cosets of the elements of B form a basis of $V(\lambda)$. $\square$

5 Complexity issues

In this section we let $\lambda_1, \dots, \lambda_l$ denote the fundamental weights corresponding to the root system Φ. We recall that any dominant weight λ can be written as $\lambda = p_1\lambda_1 + \cdots + p_l\lambda_l$, where the p_i are non-negative integers. The input to the algorithm HighestWeightModule consists of the vector $(p_1, \dots, p_l)$ together with the multiplication table of L, (1). We note that the algorithm is entirely rational, i.e., the scalars appearing are all from $\mathbb{Q}$, regardless of the ground field of L. For this reason we assume that the field we are working in is $\mathbb{Q}$.

By the complexity of an algorithm we mean a function describing the number of "primitive operations" executed by the algorithm in terms of the size of the input. We explain what is meant by "size" of an object. The size of an integer is its number of digits in binary representation. We take any object to be represented by an integer or a list of integers. The size of any object is the sum of the sizes of its constituents, e.g., the size of a rational number p/q (where $\gcd(p, q) = 1$) is equal to the sum of the sizes of p and q. Usually the term "primitive operation" either means "arithmetical operation" or "bit operation". Here we take bit operations as primitive operations, because the number of bit operations is a more realistic measure of the "work" performed by a computer program.

An algorithm is said to run in polynomial time if its complexity is bounded by a polynomial in the size of the input. We note that there are polynomial time algorithms for performing the standard arithmetical operations in the field $\mathbb{Q}$. Therefore an algorithm that performs a polynomial number of arithmetical operations and works with objects of polynomial size, has a polynomial bit complexity.

Let s be the number of positive roots of Φ. We use s as a parameter describing the size of the multiplication table of L. This is reasonable since the number of non-zero structure constants is quadratic in s; furthermore the structure constants are small integers that do not grow with L, cf. [14], §25.2. Now Weyl's dimension formula reads

$$\dim V(\lambda) = \prod_{\alpha>0} \frac{(\alpha, \lambda+\rho)}{(\alpha, \rho)},$$

where $\rho = \lambda_1 + \cdots + \lambda_l$. From this it follows that $\dim V(\lambda) \geq C(p_{\min}+1)^s$, where $p_{\min}$ is the smallest of the coefficients p_i. And C is a coefficient depending on the Lie algebra L. So no algorithm for finding a basis of $V(\lambda)$ will have a polynomial complexity in both λ and s. Therefore we consider the complexity of computing $V(\lambda)$ where L is fixed and λ varies. This means that the input to the algorithm is λ, given by the list of coefficients p_i. Again by Weyl's dimension formula we see that $\dim V(\lambda)$ is polynomial in the coefficients of λ. First we consider the complexity of multiplying two monomials in $U(L)$.

Lemma 17 *Let a, a' be two monomials in $U(L)$ of the form (2), with exponents n_α, k_i, m_α and $n'_\alpha, k'_i, m'_\alpha$ respectively. Then the complexity of computing the product $a \cdot a'$ (as a linear combination of monomials) using the multiplication law of $U(L)$ (cf. Section 2) is polynomial in the exponents of the monomials a, a'.*

PROOF. The multiplication law of $U(L)$ is obtained by a number of applications of the formulas of Lemmas 1, 2, 3, 4 and formulas (6), and (7). Furthermore the number of such applications is fixed as the Lie algebra L is fixed. By inspecting these formulas we see that every application of a formula results in a number of terms that is polynomial in the exponents of a and a'. The coefficients appearing in these formulas are of the form c^n and $\binom{n+m}{k}$. The sizes of both these numbers are polynomial in m, n, k (the size of c^n is n times the size of c, and the size of $n!$ is bounded by n times the size of n). Therefore applying the multiplication law of $U(L)$ to $a \cdot a'$ results in a polynomial number of terms, with coefficients of polynomial size. □

Remark 18 *The multiplication in $U(L)$ is in general not polynomial in s. To see this let L be a Lie algebra of type A_l. We denote the simple roots of L by $\alpha_1, \ldots, \alpha_l$. Then the positive roots of L are $\alpha_i+\alpha_{i+1}+\cdots+\alpha_{i+j}$, where $1 \leq i \leq l$ and $0 \leq j \leq l-i$. As in Section 3 we write $y_i = x_{-\alpha_i}$, where $\alpha_1, \ldots, \alpha_s$ are the positive roots. Then there are $l-1$ elements y_k with which y_1 does not commute, namely the y_k corresponding to the roots $\alpha_2, \alpha_2+\alpha_3, \ldots, \alpha_2+\cdots+\alpha_l$. We denote these y_k by $y_{k_1}, \ldots, y_{k_{l-1}}$. Now if "move" $y_1^{(m_1)}$ to the front in the expression*

$$y_1^{(n_1)} \cdots y_s^{(n_s)} y_1^{(m_1)} \cdots y_s^{(m_s)},$$

then we have to apply formula II of Lemma 1 $l-1$ times. This results in an $(l-1)$-fold summation, where the summation indices $i_1, \ldots, i_{l-1}$ take all possible values such that $i_j \geq 0$, $i_j \leq n_{k_j}$ and $m_1 - i_1 - \cdots - i_{l-1} \geq 0$. Taking *the n_{k_j} and m_1 all equal to r we see that after moving $y_1^{(m_1)}$ to the front we already have $O(r^{l-1})$ terms, which is exponential in l and hence in s.*

Lemma 19 *Let m be the maximum of $\lambda(h_{\alpha_i})$ where i runs from 1 to s. Then the size of the union of all sets M_μ constructed in Step 2a of the algorithm* HighestWeightModule *(i.e., after the erasing operation), is bounded by $(m+2)^s$.*

PROOF. Fix an index $1 \leq i \leq s$, and set $h_i = h_{\alpha_i}$. Furthermore, set $x_i = x_{\alpha_i}$ and $y_i = y_{\alpha_i}$. Then h_i, x_i, y_i span a subalgebra K_i of L that is isomorphic to $\mathfrak{sl}_2(F)$. Let 1 denote the image of $1 \in A(\lambda)$ in $V(\lambda)$. Then the elements $y_i^{(k)} \cdot 1$ for $k \geq 0$ span an irreducible K_i-submodule U_i of $V(\lambda)$. Now by the representation theory of $\mathfrak{sl}_2(F)$, we have that $\dim U_i = d_i + 1$, where d_i is such that $h_i \cdot 1 = d_i \cdot 1$ (cf. [14], §7.2). But this means that $d_i = \lambda(h_i)$. We conclude that $y_i^{(\lambda(h_i)+1)} \cdot 1 = 0$, i.e., $y_i^{(\lambda(h_i)+1)} \in I(\lambda)$, but $y_i^{(k)} \notin I(\lambda)$ for $1 \leq k \leq \lambda(h_i)$.

For $1 \leq i \leq s$ we set $\mu_i = \lambda - (\lambda(h_i) + 1)\alpha_i$. Then by the above $\mu_i \in D$ (as $\mu_i + \alpha_i$ is a weight of $V(\lambda)$), but μ_i is not a weight of $V(\lambda)$. Now let $\mu \in D$ and let $a = y_1^{(n_1)} \cdots y_s^{(n_s)}$ be a monomial of weight μ considered in Step 2a of the algorithm. Suppose that there is an index i such that $n_i \geq \lambda(h_i) + 1$. Then we consider two possibilities. First suppose that $\mu \neq \mu_i$. Then the level of μ is strictly bigger than the level of μ_i. So as $\mu_i \in D$, this weight has already been dealt with in the algorithm. Therefore, by the conclusion of the first part of the proof, we have that $y_i^{(\lambda(h_i)+1)}$ is already contained in the Gröbner basis. Therefore a is divisible by the leading monomial of an element of the Gröbner basis, and hence will be erased from the set M_μ. Secondly suppose that $\mu = \mu_i$, then we have $a = y_i^{(\lambda(h_i)+1)}$, and this element is not divided by the leading monomial of any element of the Gröbner basis. Hence a is added to the Gröbner basis.

It follows that if $a = y_1^{(n_1)} \cdots y_s^{(n_s)}$ is a monomial surviving the erasing operation of Step 2a, then $0 \leq n_i \leq \lambda(h_i)$, or $a = y_i^{(\lambda(h_i)+1)}$. In particular we see that $0 \leq n_i \leq \lambda(h_i) + 1$, and the number of all such monomials is bounded by $(m+2)^s$. □

Corollary 20 *Let m be as in Lemma 19. Let G be the Gröbner basis output by the algorithm* HighestWeightModule*. Then the number of elements G is bounded by $(m+2)^s$.*

PROOF. This immediately follows from Lemma 19, as every element from G has a unique leading monomial coming from the sets of monomials M_μ constructed in Step 2a of the algorithm. □

Lemma 21 *Let G be the Gröbner basis computed by the algorithm* HighestWeightModule*, and let $g \in G$. Then the sizes of the coefficients of the monomials in g are bounded by a polynomial in the coefficients of the highest weight λ.*

PROOF. As before we write $\lambda = p_1\lambda_1 + \cdots + p_l\lambda_l$. Let D be the extended weight diagram of λ. Let $\mu \in D$ and let g be an element of G of weight μ. Furthermore, let $M_\mu = \{a_1, \ldots, a_t\}$ be the set of monomials of weight μ constructed in Step 2a of the algorithm (i.e., as it is after the erasing operation). Write $x_1, \ldots, x_s$ for $x_{\alpha_1}, \ldots, x_{\alpha_s}$. If a is a monomial in $A(\lambda)$ of the form $a = y_1^{(n_1)} \cdots y_s^{(n_s)}$, then we set $a' = x_1^{(n_1)} \cdots x_s^{(n_s)}$. Such an a' acts on the left on $A(\lambda)$. Furthermore, for $a_i, a_j \in M_\mu$ we have that $a_i' \cdot a_j = e_{ij} \cdot 1$, where e_{ij} is an integer. Let E be the matrix $(e_{ij})_{1 \leq i,j \leq t}$. First we study the sizes of the entries e_{ij}. For that we work inside the algebra $U(L)$. We write the product $a_i'a_j$ as a linear combination of monomials of the form (2). From this linear combination we take all monomials that are in $U(H)$, with their coefficients. Denote this expression by h_{ij}. In h_{ij} we substitute $h_i \to \lambda(h_i) = p_i$ and after evaluating we get e_{ij}. Now h_{ij} is a linear combination of elements of the form

$$\binom{h_1}{k_1} \cdots \binom{h_l}{k_l}. \tag{9}$$

Furthermore, the sizes of the coefficients of the elements (9) in h_{ij} are polynomial in the sizes of the exponents n_i by Lemma 17. As seen in the proof of Lemma 19, the exponents n_i are bounded by a polynomial in the coefficients of λ. Hence the sizes of the coefficients appearing in the h_{ij} are polynomial in the size of λ. Now every monomial (9) such that there is an index i with $k_i > p_i$ is zero after the substitution. We delete those monomials from h_{ij}. Then h_{ij} is a polynomial in the $h_1, \ldots, h_l$ of degree at most $p_1 \cdots p_l$. So the sizes of the entries e_{ij} are polynomial in the size of λ.

Write $g \in G$ of weight μ as $g = \sum g_i a_i$. Now $a_j' \cdot g$ is an element of weight λ. Also by Proposition 9 we have that $a_j' \cdot g \in I(\lambda)$. But the only element of weight λ contained in $I(\lambda)$ is 0, i.e., $a_j' \cdot g = 0$ in $A(\lambda)$. Hence the vector (g_i) must be a solution of the equation $xE = 0$. Conversely, any solution to this equation gives rise to an element of $I(\lambda)$ of weight μ. A set of solutions to $xE = 0$ can be computed in polynomial time, such that the sizes of the solutions are polynomial in the coefficients of E (see, e.g., [9]). After (maybe) reducing these solutions modulo each other, and dividing each element by the coefficient of its leading monomial, we get the elements of G of weight μ. Hence the sizes of the coefficients of the elements in G are polynomial in the size of λ. $\square$

Corollary 22 *Let L be a semisimple Lie algebra. Then the complexity of the* *algorithm* HighestWeightModule *is polynomial in the size of the highest weight* λ.

PROOF. From Corollary 20 we have that the number of elements of the Gröbner basis is polynomial in the size of λ. Furthermore Lemma 19 implies that the number of monomials occurring in each element of the Gröbner basis is polynomial. Hence in Step 2c a polynomial number of S-elements are constructed. Furthermore, they are reduced modulo a set of polynomial size. So the number of multiplications of two monomials in $U(N^-)$ that is carried out by the algorithm is polynomial in the size of λ. As seen in the proof of Lemma 19, the exponents of these monomials are bounded by a polynomial in λ. Also by Lemma 21 the coefficients of the monomials of the elements of the Gröbner basis are of polynomial size. Hence Lemma 17 implies that the loop of Step 2c takes a polynomial number of bit operations. □

Remark 23 *Some of the results of this section suggest that the algorithm is exponential in the number of positive roots* s *(the multiplication in* $A(\lambda)$ *is exponential in* s *for example). In the next section some examples are presented that indicate that this is indeed the case.*

6 Example and practical experiences

In this section we construct the 7-dimensional irreducible module over the Lie algebra of type G_2. A Cartan matrix of G_2 is

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

There are six positive roots, namely α_1, α_2, $\alpha_1+\alpha_2$, $2\alpha_1+\alpha_2$, $3\alpha_1+\alpha_2$ and $3\alpha_1+2\alpha_2$, where α_1, α_2 are the simple roots. Corresponding to these roots there are the negative root vectors, which we denote by $y_1, \dots, y_6$. It is possible to choose a multiplication table of L such that the y_i satisfy the commutation relations

$$[y_1, y_2] = y_3,\ [y_1, y_3] = 2y_4,\ [y_1, y_4] = 3y_5,\ [y_2, y_5] = y_6,\ [y_3, y_4] = -3y_6$$

and all other products are zero.

We denote the fundamental weights by λ_1, λ_2. In terms of the fundamental weights the roots are given by $\alpha_1 = 2\lambda_1 - \lambda_2$ and $\alpha_2 = -3\lambda_1 + 2\lambda_2$. In the sequel we denote a weight $p_1\lambda_1 + p_2\lambda_2$ by (p_1, p_2). The highest weight of the 7-dimensional irreducible module over G_2 is $(1,0)$. The extended weight diagram is displayed in Figure 1. The weights are listed according to increasing level. Let μ be a weight in this diagram, occurring on line k (so its level is $k-1$). Let ν be a weight just below μ. If it occurs to the left of μ, then $\nu = \mu - \alpha_1$. If it occurs to the right of μ, then $\nu = \mu - \alpha_2$. The weights that are weights of the module are underlined. The dimension of all weight spaces is one. (This can be calculated using a standard algorithm, cf. [4], [13], [18]).

$$\begin{array}{c}
\underline{(1,0)} \\[1ex]
\underline{(-1,1)} \quad (4,-2) \\[1ex]
(-3,2) \quad \underline{(2,-1)} \\[1ex]
\underline{(0,0)} \qquad (5,-3) \\[1ex]
\underline{(-2,1)} \quad (3,-2) \\[1ex]
(-4,2) \quad \underline{(1,-1)} \\[1ex]
\underline{(-1,0)} \quad (4,-3) \\[1ex]
(-3,1) \quad (2,-2) \\[1ex]
(-5,2) \quad (0,-1) \\[1ex]
(-2,0) \\[1ex]
(-4,1) \quad (1,-2) \\[1ex]
(-1,-1)
\end{array}$$

Fig. 1. Extended weight diagram of the highest-weight module over the Lie algebra of type G_2, with highest weight $(1,0)$.

Now going through the extended weight diagram, we construct a Gröbner basis of the ideal $I(1,0)$. The dimension of the weight space with weight $(1,0)$ is one, and we denote a basis vector of it by 1. Then $y_1 \cdot 1 = y_1$ is a weight vector of weight $(-1,1)$. So it is a basis vector of that weight space. Furthermore, y_2 is a basis vector of the weight space with weight $(4,-2)$. But this weight is not a weight of the representation, so y_2 is an element of the Gröbner basis. Now we consider the weight $(-3,2)$. The monomial $y_1^{(2)}$ is in the corresponding weight space, but as $(-3,2)$ is not a weight of the representation we have that $y_1^{(2)}$ is an element of the Gröbner basis. There are two monomials of weight $(2,-1)$, namely y_1y_2 and y_3. However, the first of these reduces to zero modulo y_2, which is an element of the Gröbner basis. So only y_3 remains. We consider the weight $(0,0)$. Again there are two monomials of this weight, y_1y_3 and y_4, none of which reduce modulo the Gröbner basis computed so far. However, the S-element $S(y_1^{(2)}, y_2)$ is of weight $(0,0)$. It equals $-y_1y_3 + y_4$. So we add it to the Gröbner basis, and only one basis vector, y_4 remains. Continuing like this we take care of all weights of the diagram, and end up with the Gröbner basis we are looking for. It consists of the elements

$$\begin{gathered}y_2,\ y_1^{(2)},\ y_1y_3 - y_4,\ y_1y_4 - 2y_5,\ y_3^{(2)},\ y_1y_5,\ y_3y_4 + 2y_6,\\ y_3y_5 - y_1y_6,\ y_4^{(2)} + y_1y_6,\ y_3y_6,\ y_4y_5,\ y_4y_6,\ y_5^{(2)},\ y_5y_6,\ y_6^{(2)}.\end{gathered}$$

A basis of the module is formed by the elements $1,\ y_1,\ y_3,\ y_4,\ y_5,\ y_6,\ y_1y_6$. The action of the elements of L is calculated by using the multiplication in $A(1,0)$ together with the Gröbner basis. Let $x_1, \dots, x_6$ be the elements of the Chevalley basis corresponding to the positive roots. We calculate the action of x_3 on y_1y_6. First we have that x_3 times y_1y_6 equals $-y_1y_4 + y_1y_6x_3 + 3y_5 + 3y_6x_2$. But $y_1y_6x_3$ and $3y_6x_2$ are zero in $A(1,0)$. Furthermore, $-y_1y_4$ reduces modulo the Gröbner basis to $-2y_5$. So x_3 maps the vector y_1y_6 to y_5.

I have implemented the algorithm HighestWeightModule in the computer algebra system GAP4 ([11]). The implementation is part of the latest release (GAP4.2) of this system.

Some timings ([1]) are displayed in Tables 1 and 2. From these tables we see that the algorithm performs well enough to be able to compute modules of dimensions into the thousands. However, as suggested by the complexity analysis of Section 5, the running time is rather sensitive to the number of positive roots of the input Lie algebra. Both for F_4 and E_7 we have calculated a module of dimension roughly 8500. But the running time in the case of E_7 is markedly longer than the one for F_4.

To further investigate the dependence of the running time of the algorithm on the number of positive roots we have calculated the adjoint module of the Lie algebras of type C_l for $l = 7, \dots, 11$. The results are displayed in Table 3. From this table the running times appear to be exponential in l (they roughly double each step). However, the size of the Gröbner basis does not seem to grow as quickly as the running time. This suggests that the exponential behaviour of the running time is mainly caused by the increasing complexity of the multiplication in $U(L)$.

[1] The computations were done on a Pentium 266.

λ	$\dim V(\lambda)$	time (s)	size of G
(0,0,1,0)	273	10	458
(1,0,0,1)	1053	35	940
(0,1,0,0)	1274	44	1094
(0,0,1,1)	4096	209	2786
(1,0,1,0)	8424	448	4041
(0,1,0,1)	19278	1910	7197

Table 1
Running times for some irreducible representations of F_4. The fourth column displays the size of the Gröbner basis. (The numbering of the simple roots is as in [5].)

λ	$\dim V(\lambda)$	time (s)	size of G
(1,0,0,0,0,0,0)	133	32	591
(0,1,0,0,0,0,0)	912	230	2177
(0,0,0,0,0,1,0)	1539	378	3171
(1,0,0,0,0,0,1)	6480	2040	7347
(0,0,1,0,0,0,0)	8645	3504	10182

Table 2
Running times for some irreducible representations of E_7. The fourth column displays the size of the Gröbner basis. (The numbering of the simple roots is as in [5].)

l	$\dim \mathrm{ad}\, C_l$	time (s)	size of G
7	105	16	491
8	136	36	729
9	171	72	1033
10	210	146	1411
11	253	281	1871

Table 3
Running times for the calculation of the adjoint module of C_l for $7 \leq l \leq 11$. The fourth column displays the size of the Gröbner basis.

References

[1] J. Apel and W. Lassner. An extension of Buchberger's algorithm and calculations in enveloping fields of Lie algebras. *J. Symbolic Comput.*, 6(2-3):361–370, 1988.

[2] G. E. Baird and L. C. Biedenharn. On the representations of the semisimple Lie groups. II. *J. Mathematical Phys.*, 4:1449–1466, 1963.

[3] A. O. Barut and R. Raczka. *Theory of group representations and applications.* World Scientific Publishing Co., Singapore, second edition, 1986.

[4] R. E. Beck and B. Kolman. Computers in Lie algebras. I. Calculation of inner multiplicities. *SIAM J. Appl. Math.*, 25:300–312, 1973.

[5] N. Bourbaki. *Groupes et Algèbres de Lie, Chapitres 4, 5 et 6.* Hermann, Paris, 1968.

[6] N. Bourbaki. *Groupes et Algèbres de Lie, Chapitres VII et VIII.* Hermann, Paris, 1975.

[7] R. W. Carter. *Simple groups of Lie type.* John Wiley & Sons, London-New York-Sydney, 1972. Pure and Applied Mathematics, Vol. 28.

[8] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra.* Springer Verlag, New York, Heidelberg, Berlin, 1992.

[9] M. A. Frumkin. Polynomial time algorithms in the theory of linear Diophantine equations. pages 386–392. Lecture Notes in Comput. Sci., Vol. 56, 1977.

[10] W. Fulton and J. Harris. *Representation Theory.* Springer Verlag, New York, Heidelberg, Berlin, 1991.

[11] The GAP Group, Aachen, St Andrews. *GAP – Groups, Algorithms, and Programming, Version 4.2*, 2000. (`http://www-gap.dcs.st-and.ac.uk/~gap`).

[12] I. M. Gel'fand and M. L. Cetlin. Finite-dimensional representations of the group of unimodular matrices. *Doklady Akad. Nauk SSSR (N.S.)*, 71:825–828, 1950. (in Russian).

[13] W. A. de Graaf. *Lie Algebras: Theory and Algorithms*, volume 56 of *North-Holland Mathematical Library.* Elsevier Science, 2000.

[14] J. E. Humphreys. *Introduction to Lie Algebras and Representation Theory.* Springer Verlag, New York, Heidelberg, Berlin, 1972.

[15] N. Jacobson. *Lie Algebras.* Dover, New York, 1979.

[16] A. Kandri-Rody and V. Weispfenning. Noncommutative Gröbner bases in algebras of solvable type. *J. Symbolic Comput.*, 9(1):1–26, 1990.

[17] P. Littelmann. An algorithm to compute bases and representation matrices for sl_{n+1}-representations. *J. Pure Appl. Algebra*, 117/118:447–468, 1997.

[18] R. V. Moody and J. Patera. Fast recursion formula for weight multiplicities. *Bull. Amer. Math. Soc. (N.S.)*, 7(1):237–242, 1982.