COMPUTATIONS WITH REPRESENTATIONS OF LIE GROUPS

Bart de Smit

Center for Mathematics and Computer Science
Amsterdam, The Netherlands

June 1989

COMPUTATIONS WITH REPRESENTATIONS OF LIE GROUPS

Bart de Smit

Center for Mathematics and Computer Science
Amsterdam, The Netherlands
June 1989

Introduction

Since January 1988 a software package called **LiE** is under development at CWI in Amsterdam. Its purpose is to enable mathematicians and physicists to obtain on-line information as well as to interactively perform computations of a Lie group theoretic nature. Our primary goal in realising the present version has been to cover the mathematical content of the books by Bremner, Moody & Patera [5], McKay & Patera [14] and Tits [20]. The mathematical programs were written by Arjeh Cohen, Ron Sommeling and Bart de Smit, the interpreter was written by Bert Lisser and the package was tested by Bert Ruitenburg.

The three chapters of the present paper are intended to give an impression of what **LiE** is about, how it can be used, and what mathematical ideas have been used in the program. In the first chapter an overview is given of the mathematical theories that **LiE** is based upon. There are several points of view, which all result in the more or less combinatorial context of root systems. The way this is done is briefly described for Lie algebras, Lie groups and algebraic groups. The second chapter contains an introduction to the use of the package. The main routines that can be called in **LiE** are explained in terms of the terminology of the first chapter. Finally, a more detailed description is given in the third chapter of some of the key algorithms used in the program. In particular it is shown how Weyl group orbits of weights can be enumerated.

The package was written in C with the help of "lex" and "yacc", so it should run in any UNIX-like environment. At CWI it runs on a VAX 11/780 (BSD UNIX 4.3), a SUN 4 (SunOS 4.0) and an IBM RT (AIX). For more details see the **LiE** manual [13], or contact the Computer Algebra Group, c/o Mrs W. van Eijk, CWI, P.O. Box 4079, 1009 AB Amsterdam.

Chapter 1. Lie algebras, Lie groups and algebraic groups

This chapter contains a brief treatment of the mathematical theory of Lie algebras, Lie groups and algebraic groups. In particular a resumé of representation theory for each of the three cases is given. We first consider root systems, as they form an indispensable tool for studying each of these three topics and doing explicit calculations. Root systems are relatively easy to grasp finite combinatorial structures that can be represented entirely in a computer, in contrast with Lie algebras, Lie groups and algebraic groups.

1.1. Root systems

In this section a brief outline is given of the theory of root systems, Dynkin diagrams and weights. Main references: Humphreys [10], Serre [18] and Bourbaki [3].

1.1.1 Definition of root systems. A *euclidean space* E is a finite dimensional real vector space with a non-degenerate positive definite symmetric bilinear form $(\cdot,\cdot)$. For non-zero $\alpha \in E$ the reflection σ_α in the hyperplane perpendicular to α is given by

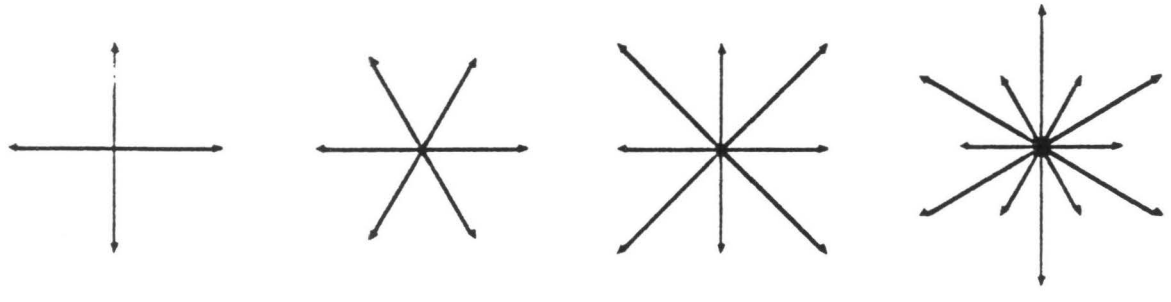
$$\sigma_\alpha : \beta \mapsto \beta - \langle \beta, \alpha \rangle \alpha,$$

where $\langle \beta, \alpha \rangle = 2(\beta, \alpha)/(\alpha, \alpha)$. A *root system* Φ in E is a finite subset of $E \backslash \{0\}$, that spans E as a vector space over $\mathbf{R}$, and that satisfies

$$\begin{array}{lll} (1) & \forall \alpha \in \Phi : & \mathbf{R}\alpha \cap \Phi = \{\alpha, -\alpha\} \\ (2) & \forall \alpha \in \Phi : & \sigma_\alpha[\Phi] = \Phi \\ (3) & \forall \alpha, \beta \in \Phi : & \langle \beta, \alpha \rangle \in \mathbf{Z} \end{array}$$

The elements of a root system are called *roots* and the dimension of the spanned euclidean space is the *rank* of the root system.

We say that two root systems Φ and Φ' in euclidean spaces E and E' are *isomorphic* if there is an isomorphism of vector spaces $\phi : E \xrightarrow{\sim} E'$ such that $\phi[\Phi] \subset \Phi'$ and $\langle \phi(\alpha), \phi(\beta) \rangle = \langle \alpha, \beta \rangle$ for all $\alpha, \beta \in \Phi$. Up to isomorphism there is only one root system, called A1 of rank 1, and it has two roots. These are all root systems of rank 2:



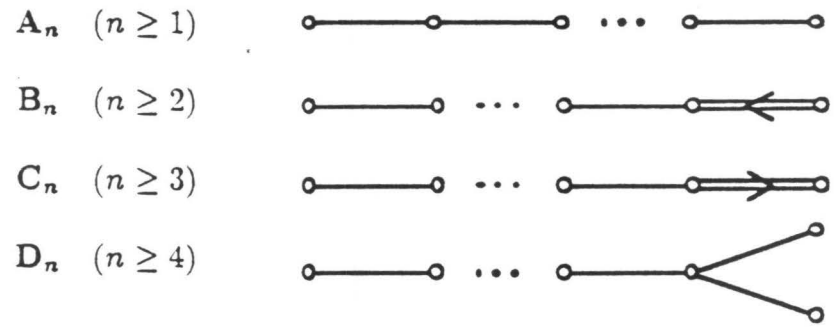
A1A1 A2 B2 G2

The *Weyl group* $\mathcal{W}$ of a root system Φ is the subgroup of $GL(E)$ generated by all σ_α with $\alpha \in \Phi$. It is easy to prove that the elements of $\mathcal{W}$ are automorphisms of Φ, and orthogonal automorphisms (isometries) of E. We may also consider $\mathcal{W}$ to be a subgroup of the permutation group of Φ, as Φ spans E. Obviously, isomorphic root systems have isomorphic Weyl groups.

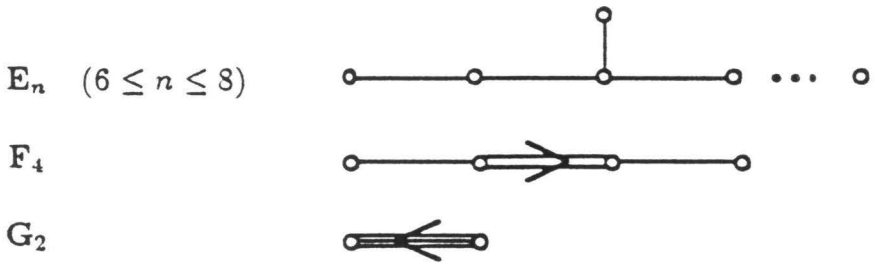
1.1.2 Dynkin diagrams. A subset Δ of Φ is called a *base* of a root system Φ in a euclidean space E if it is a base of E as a vector space over $\mathbf{R}$ and each root $\beta \in \Phi$ can be written as $\beta = \sum_{\alpha\in\Delta} k_\alpha \alpha$ with integral coefficients k_α all positive or all negative. The roots that can be written with positive coefficients are the *positive roots* (with respect to Δ). Furthermore, one can prove that every root system has a base, which is unique up to transformation by an element of the Weyl group.

Having chosen a base Δ of Φ, we construct the so-called *Dynkin diagram* of Φ as follows. It is easily shown that for $\alpha, \beta \in \Delta$ we have $\langle\alpha,\beta\rangle = \langle\beta,\alpha\rangle = 0$, or $-1 \in \{\langle\alpha,\beta\rangle, \langle\beta,\alpha\rangle\} \subset \{-1,-2,-3\}$. The Dynkin diagram is a directed graph with Δ as its set of vertices and for each pair distinct $\alpha, \beta \in \Delta$ it has $\max\{|\langle\alpha,\beta\rangle|, |\langle\beta,\alpha\rangle|\}$ edges between α and β. In this number is greater then 1, then we direct the edges from α to β if $\langle\alpha,\beta\rangle < -1$ and from β to α if $\langle\beta,\alpha\rangle < -1$. Note that the Dynkin diagrams of all bases of a root system Φ are isomorphic graphs, so that we may speak of the Dynkin diagram of Φ. A root system is determined uniquely (up to isomorphism) by its Dynkin diagram.

1.1.3 Classification of root systems. A root system is said to be *irreducible* if its Dynkin diagram is connected. Any root system can be decomposed as the union of a number of orthogonal subsets, each of which is an irreducible root system in the euclidean subspace it spans. Its Dynkin diagram then is the disjoint union of the diagrams of its irreducible components. The classification theorem of root systems gives a complete list of all possible Dynkin diagrams of irreducible root systems. See Humphreys [10], Winter [24] or Serre [18] for a proof. We have the classical root systems:



Finally, there are five exceptional root systems, E_6, E_7, E_8, F_4 and G_2:



1.1.4 The weight lattice. The $\mathbf{Z}$-span R of a root system Φ in a euclidean space E is called the *root lattice*. The lattice Λ in E consisting of all $\lambda \in E$ for which $\langle \lambda, \alpha \rangle \in \mathbf{Z}$ for all $\alpha \in \Delta$ is called the *weight lattice* of Φ. It follows from condition (3) in the definition of root systems that $R \subset \Lambda$. *Weights* are elements of Λ.

Let $\Delta = \{\alpha_1, \ldots, \alpha_r\}$ be a base of Φ, then $\alpha_1, \ldots, \alpha_r$ is also a basis of R over $\mathbf{Z}$. Now let $\lambda_1, \ldots \lambda_r \in \Lambda$ be defined by $\langle \lambda_i, \alpha_j \rangle = \delta_{ij}$, then $\lambda_1, \ldots \lambda_r \in \Lambda$ is a basis of Λ, called the basis of *fundamental weights* of Λ (with respect to Δ). The partial order $\prec$ on Λ is defined as follows: we have $\lambda \prec \mu$ if and only if $\mu - \lambda$ is a sum of positive roots. A weight $\lambda \in \Lambda$ is said to be *dominant* if for every $\alpha \in \Delta$ we have $\langle \lambda, \alpha \rangle \geq 0$ or, equivalently, if $\lambda = \sum a_i \lambda_i$ with $a_i \in \mathbf{Z}_{\geq 0}$. For every weight $\lambda \in \Lambda$ there exists an element $\sigma \in \mathcal{W}$ such that $\sigma(\lambda)$ is dominant. It is easy to see that this dominant representative of the $\mathcal{W}$-orbit of λ is unique, because it is the largest element of this orbit with respect to the ordering $\prec$.

1.2. Lie algebras

In this section a summary is given of the theory of semisimple Lie algebras over an algebraically closed field of characteristic zero. Furthermore, it is shown how a representation is given by weights. Main references: Humphreys [10], Serre [18] and Winter [24].

1.2.1 Lie algebras and modules. Let F be a field. A *Lie algebra* over F is a vector space L over F with a bilinear operation $[\cdot, \cdot] : L \times L \to L$ satisfying

$$\begin{array}{lll} \forall x, y \in L: & [x, y] = -[y, x] & (\text{anti} - \text{commutativity}) \\ \forall x, y, z \in L: & [x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0 & (\text{Jacobi} - \text{identity}). \end{array}$$

An important example is the Lie algebra $\mathfrak{gl}(V)$ of an F-vector space V: it is just the vector space of endomorphisms of V with $[x, y] = x \circ y - y \circ x$. A *homomorphism* of Lie algebras over F is a homomorphism of F-vector spaces preserving the bracket operation; a *subalgebra* of a Lie algebra is a subspace which is closed under $[\cdot, \cdot]$; an *ideal* I of a Lie algebra L is a subspace with $[I, L] \subset I$. The *centre* $Z(L)$ of a Lie-algebra L is defined to be $\{x \in L : \forall y \in L : [x, y] = 0\}$. The centre of $\mathfrak{gl}(V)$ is just the subspace of scalar matrices.

Let L be a Lie algebra over a field F. An *L-module* is a vector space V over F with a bilinear map $L \times V \to V$, denoted as $(x, v) \to x \cdot v$, such that $[x, y] \cdot v = x \cdot y \cdot v - y \cdot x \cdot v$ for all $x, y \in L$ and $v \in V$. This condition just means that the induced F-vector space homomorphism $L \to \mathfrak{gl}(V)$ is a homomorphism of Lie algebras. This homomorphism is then called the *representation* of L on V. In particular, L is an L-module and this gives rise to the *adjoint representation* $\mathrm{ad} : L \to \mathfrak{gl}(L)$, which is defined by $\mathrm{ad}(x)(y) = [x, y]$. Note that the kernel of the adjoint representation of L is the centre $Z(L)$.

A subspace W of an L-module V is said to be a *submodule* of V if $x \cdot w \in W$ for all $w \in W$. Note that a submodule of L is just an ideal of L. We say that V is a *simple* or *irreducible* module if it has precisely two submodules ($\{0\}$ and V), so in particular $\{0\}$ is not simple. A module V is said to be *semisimple* or *completely reducible* if it is the direct sum of simple modules, or equivalently if for every submodule W of V there is a submodule W' of V with $W \oplus W' = V$ (see Winter [24], Chapter 1]). We say that L is *simple* if $L \neq Z(L)$ and L is simple as a module over itself.

1.2.2 Semisimple and reductive Lie algebras. A *semisimple Lie algebra* is a finite-dimensional Lie algebra all of whose finite-dimensional modules are semisimple. A theorem of Weyl gives a necessary and sufficient condition for a Lie algebra to be semisimple. A Lie algebra L is *solvable* if $L^i = 0$ for some i, where $L^0 = L$ and $L^n = [L^{n-1}, L^{n-1}]$ for $n \in \mathbb{Z}_{>0}$. It is easily seen that every Lie algebra L has a unique maximal solvable ideal, which is called the *radical* $\mathrm{Rad}(L)$ of L. Weyl's theorem reads: let F be an algebraically closed field of characteristic zero and let L be a finite dimensional Lie algebra over F; then L is semisimple if $\mathrm{Rad}(L) = \{0\}$. Assume that F is algebraically closed and that $\mathrm{char}F = 0$. Using Weyl's theorem, one can prove that a finite dimensional Lie algebra L over F is a semisimple module over itself if and only if $\mathrm{Rad}(L) = Z(L)$. Such Lie algebras are said to be *reductive*. Again by Weyl's theorem it is easy to see that for every reductive Lie algebra L we have $L = Z(L) \oplus [L, L]$ with $[L, L]$ semisimple. From this one can deduce that the converse of Weyl's theorem holds.

1.2.3 The root system of a semisimple Lie algebra. Let F be an algebraically closed field with $\mathrm{char}F = 0$ and let L be a semisimple Lie algebra over F. As $Z(L) = \{0\}$ the adjoint representation $\mathrm{ad} : L \to \mathfrak{gl}(L)$ is injective. A subalgebra of L is said to be *toral* if its image in $\mathfrak{gl}(L)$ consists of diagonalizable endomorphisms of L. Let T be a maximal toral subalgebra of L. One can show that $[T, T] = \{0\}$ and that the image of T under any finite dimensional representation $L \to \mathfrak{gl}(V)$ consists solely of diagonalizable elements. This implies that the vector space V can be decomposed as a direct sum of subspaces as follows:

$$V = \bigoplus_{\lambda \in T^*} V_\lambda, \qquad V_\lambda = \{v \in V : \forall t \in T : t \cdot v = \lambda(t)v\},$$

where we have written T^* for the *dual* $\mathrm{Hom}_F(T, F)$ of T. The *weights* of V (with respect to T) are defined to be those $\lambda \in T^*$ for which $V_\lambda \neq \{0\}$. The space V_λ is called the *weight space* in V of weight λ. The set of *roots* Φ of L (with respect to T) is defined as the set of non-zero weights of the adjoint representation.

See Humphreys [10] for a proof that the *Killing form* $(x,y) \mapsto \mathrm{Tr}(\mathrm{ad}(x)\mathrm{ad}(y))$ on L induces a non-degenerate symmetric bilinear form on T. This gives rise to a non-degenerate symmetric bilinear form $(\cdot,\cdot)$ on T^*. At this point one proves that $\Phi \subset T^*$ satisfies all three conditions of the definition of root systems. It is then easy to deduce that the vector space $E_{\mathbf{Q}}$ over $\mathbf{Q}$ spanned by Φ has the same dimension as T over F, and that we have a non-degenerate symmetric positive definite bilinear form on $E_{\mathbf{Q}}$. It turns out that Φ is a root system in the euclidean space $\mathbf{R} \otimes_{\mathbf{Q}} E_{\mathbf{Q}}$, and that the weights of a representation are contained in the weight lattice of Φ.

1.2.4 Classification of reductive Lie algebras. The above construction of a root system out of a semisimple Lie algebra and the weight system of a representation depends on the choice of the torus T, but it can be shown that T is unique up to a so-called inner automorphism of the Lie algebra L. See Winter [24] or Humphreys [10] for a Lie algebra theoretic proof, and Humphreys [11] for a proof using algebraic geometry. A Lie algebra isomorphism $\sigma : L \xrightarrow{\sim} L'$ induces an isomorphism between the root system of L with respect to T and the root system of L' with respect to $\sigma[T]$, and conversely, semisimple Lie algebras over F with isomorphic root systems are isomorphic as Lie algebras. Finally, a straightforward construction of Serre shows that every isomorphism class of root systems can be obtained in this way (see the appendix of Serre [18]). Thus the semisimple Lie algebras over F are classified as in the previous section. The simple Lie algebras correspond to irreducible root systems, and every semisimple algebra is a product of simple algebras. For example, the Lie algebra $\mathfrak{sl}(n,F) = \{x \in \mathfrak{gl}(n,F) : \mathrm{Tr}(x) = 0\}$ has a root system of type $\mathbf{A}_{n-1}$. The classification can be extended to comprise all reductive Lie algebras over F: they are of the form $H \oplus L$, with L a semisimple Lie algebra, and H a Lie algebra with $[\cdot,\cdot]$ identically zero. Denoting the isomorphism class of such an algebra H of dimension n by $\mathbf{T}_n$, we see that the Lie algebra $\mathfrak{gl}(n,F) \cong F \oplus \mathfrak{sl}(n,F)$ is of type $\mathbf{T}_1\mathbf{A}_{n-1}$.

1.2.5 Representations of semisimple Lie algebras. Again, we work over an algebraically closed field F of characteristic zero. Let L be a semisimple Lie-algebra and V a finite dimensional irreducible L-module. Fix a maximal toral subalgebra T of L, and denote the root system of L with respect to T by Φ. Let Δ be a base of Φ, then the set of weights of V is partially ordered by $\prec$. It can be shown that there is a unique maximal weight λ of V, and that this weight is dominant. Conversely, there is a unique irreducible L-module $V(\lambda)$ with highest weight λ with respect to T for each dominant weight λ in the weight lattice of Φ. As every finite dimensional L-module is semisimple, this classifies the finite dimensional modules over L in terms of the root system of L: every finite dimensional module V is given as $V = \oplus V(\lambda)^{m(\lambda)}$ where the sum ranges over all dominant weights λ, and the $m(\lambda)$ are non-negative integers, almost all equal to zero, and uniquely determined by V.

To conclude this section, we look at the weight system of the irreducible module $V(\lambda)$ with highest weight λ. All weights of V are of the form $\lambda - \sum_{\alpha\in\Delta} k_i\alpha$ with $k_i \in \mathbf{Z}_{\geq 0}$. Furthermore, the set of weights $\Pi(\lambda)$ of $V(\lambda)$ is permuted by the Weyl group $\mathcal{W}$ of Φ. The dimension $m(\mu,\lambda) = \dim_F V(\lambda)_\mu$ of the weight space of $V(\lambda)$ of weight μ is called the *multiplicity* of μ in $V(\lambda)$. Two weights $\mu, \mu' \in \Pi(\lambda)$ in the same $\mathcal{W}$-orbit

have the same multiplicity, so for the calculation of multiplicities one can restrict to the case where μ is dominant. We have $m(\mu,\lambda)=0$ if $\mu \not\preceq \lambda$ and $m(\lambda,\lambda)=1$. Other multiplicities can be calculated by *Freudenthal's recursion formula*:

$$((\lambda+\delta,\lambda+\delta)-(\mu+\delta,\mu+\delta))m(\mu,\lambda)=2\sum_{\alpha\succ 0}\sum_{i=0}^{\infty} m(\mu+i\alpha,\lambda)(\mu+i\alpha,\alpha),$$

where $\delta=\frac{1}{2}\sum_{\alpha\succ 0}\alpha$, and the summations over "$\alpha\succ 0$" are taken over all positive roots α. Note that the infinite summation is actually a finite summation since we have $m(\mu+i\alpha,\lambda)=0$ for i sufficiently large. The inner product used in this formula can be changed by a scalar without affecting the outcome, so it may be chosen in any convenient way. The dimension of an irreducible module of highest weight weight λ can be calculated by *Weyl's formula*:

$$\dim_F V(\lambda)=\prod_{\alpha\succ 0}\frac{\langle\lambda+\delta,\alpha\rangle}{\langle\delta,\alpha\rangle}.$$

1.3. Lie groups

The concept of a Lie group combines the algebraic notion of groups with the analytic notion of manifolds. Lie groups occur widely in geometry, analysis and physics. As in the previous section it is shown how the classification and representation theory relates to the theory of root systems and weights. For instance, we show how the simply connected compact real Lie groups correspond bijectively (up to isomorphism) to root systems. Main references: Warner [22], Bröcker&Dieck [6] and Varadarajan [21].

1.3.1 Lie groups. A *Lie group* is an analytic manifold G with a group structure such that the multiplication map $G\times G\to G$ and the inverse map $G\to G$ sending $g\in G$ to g^{-1} are analytic. Here the word analytic means either real-analytic or complex-analytic (holomorphic), and we speak of real and complex Lie groups accordingly. The identity element of a Lie group G is denoted by e. A *homomorphism* $G\to G'$ of Lie groups (either both real or both complex) is a group homomorphism which is analytic as a map between manifolds. Important examples of Lie groups are the n-dimensional *torus* $T^n=\mathbf{R}^n/\mathbf{Z}^n$ and the groups $GL(n,\mathbf{R})$ and $GL(n,\mathbf{C})$.

The question whether every locally euclidean topological group satisfying the second countability axiom has a structure of a real Lie group is known as Hilbert's Fifth Problem. It turns out that these groups have a unique real analytic structure, and that a continuous homomorphism of such groups is a homomorphism of Lie groups with respect to this analytic structure. (In particular, a continuous group homomorphism of Lie groups is always analytic.) See Montgommery & Zippin [15] for more information.

The connected component G° of e in a Lie group G is easily seen to be an open and closed normal subgroup of finite index in G. For example, $G=GL(n,\mathbf{R})$ has

two connected components for any $n \in \mathbf{Z}_{>0}$ and G° is the subgroup of all invertible matrices of positive determinant.

1.3.2 The Lie algebra of a Lie group. Recall that the tangent space T_pM of an m-dimensional (real or complex) analytic manifold M at a point $p \in M$ is the m-dimensional (real or complex) vector space. The derivative of an analytic map $f : M \to N$ at a point $p \in M$ is a homomorphism $d_pf : T_pM \to T_{f(p)}N$ of vector spaces. Recall that tangent spaces T_pM at points p of a M can be pasted in order to obtain the tangent bundle TM of M, which is a vector bundle of dimension n over M, and an analytic map $f : M \to N$ has a analytic derivative $df : TM \to TN$. The tangent bundle of a Lie group G is always trivial (in a canonical way): left translation by an element $x \in G$ is an invertible analytic map $l_x : G \to G$ given by $g \mapsto xg$, so we have an isomorphism $d_el_x : T_eG \xrightarrow{\sim} T_x$ for each $x \in G$, which gives an isomorphism $G \times T_eG \xrightarrow{\sim} TG$ of vector bundles over G defined by $(x, v) \mapsto d_el_x(v)$. Thus a tangent vector v at e extends uniquely to a global vector field X of G that is left invariant in the sense that for all $x, y \in G$ we have $X(xy) = d_yl_x(X(y))$. The space of left invariant vector fields on G can therefore be identified with T_eG, and we call this vector space the *Lie algebra* $\mathcal{L}G$ of G. A global vector field on M is a derivation of the algebra of analytic functions on M, and defining the Lie-bracket operation $[X, Y] = X \circ Y - Y \circ X$ for global vector fields X and Y, the vector space $\mathcal{L}G$ is equipped with the structure of a Lie algebra in the sense of the preceding section.

A homomorphism $f : G \to H$ of Lie groups gives a homomorphism $df : \mathcal{L}G \to \mathcal{L}H$ of Lie algebras, namely the derivative in e. Conversely, if G and H are connected Lie groups and $\phi : \mathcal{L}G \to \mathcal{L}H$ a homomorphism of Lie algebras, then there exists at most one Lie group homomorphism $f : G \to H$ with $df = \phi$. Furthermore such an f exists if G is *simply connected*, that is, if G has a trivial fundamental group. For a proof see Varadarajan [21], theorem 2.7.5. It can be shown that the universal cover of a Lie group (in both the real and the complex case) has a canonical structure of a Lie group, so every connected Lie group is the quotient of a simply connected Lie group by a discrete normal subgroup. In fact this holds for a much wider range of topological groups; see e.g. Pontryagin [17].

For $v \in \mathcal{L}G$ it follows from the preceding paragraph that there is a unique Lie group homomorphism $\alpha_v : \mathbf{R} \to G$, with $d\alpha_v(\frac{\partial}{\partial x}) = v$, where $\frac{\partial}{\partial x} \in \mathcal{L}\mathbf{R}$ is the unit global vector field on $\mathbf{R}$. We can now define the exponential map $\exp : \mathcal{L}G \to G$ by $v \mapsto \alpha_v(1)$. The exponential map is analytic and it is functorial in the sense that for a morphism of Lie groups $f : G \to H$ with derivative $d_ef : \mathcal{L}G \to \mathcal{L}H$ we have $\exp \circ d_ef = f \circ \exp$. If $G = GL(V)$ with V a finite dimensional complex or real vector space then $\mathcal{L}G = \mathfrak{gl}(V)$ and the exponential map for G is given by the usual power series $A \mapsto \sum_{i=0}^{\infty} \frac{1}{i!}A^i$.

1.3.3 Modules over a Lie group. A (complex) *module* over a Lie group G is a complex vector space V with an analytic $\mathbf{C}$-linear group action $G \times V \to V$ (note that V has a canonical analytic structure). Again we have the notions of submodule, simple module and semisimple module. A *complex representation* of a Lie group G on

a complex vector space V is a homomorphism $G \to GL(V)$ of Lie groups. As with Lie algebras a representation of G on V is the same thing as a G-module structure on V.

For instance, if G is a complex Lie group then the *adjoint representation* Ad : $G \to GL(\mathcal{L}G)$ is defined by sending an element $x \in G$ to the derivative in e of the conjugation map $G \to G$ given by $y \mapsto xyx^{-1}$. A straight-forward computation shows that the derivative $d\mathrm{Ad} : \mathcal{L}G \to \mathcal{L}GL(\mathcal{L}G) = \mathfrak{gl}(\mathcal{L}G)$ is just the adjoint representation ad of $\mathcal{L}G$. In general the derivative of a representation $G \to GL(V)$ is a Lie algebra representation $\mathcal{L}G \to \mathfrak{gl}(V)$. If G is a real Lie group, we obtain a map $G \to GL(\mathcal{L}G)$ in the same way, but as we insist on modules being complex vector spaces, the adjoint representation for G is defined to be the composite homomorphism of real Lie groups $G \to GL(\mathcal{L}G) \to GL(\mathcal{L}G \otimes_{\mathbf{R}} \mathbf{C})$.

1.3.4 The compact torus. Denote the multiplicative group of non-zero complex numbers by $\mathbf{C}^*$, and let T^1 be the subgroup of $\mathbf{C}^*$ consisting of those $x \in \mathbf{C}^*$ with $|x| = 1$. The representation theory for the n-dimensional *torus* $T^n = (T^1)^n$, is particularly easy: every irreducible (complex) module is one-dimensional. Consequently, an irreducible representation of T^n is just a character $T^n \to \mathbf{C}^*$, so it is always of the form $(x_1, \ldots, x_n) \mapsto \prod_{i=1}^n x_i^{a_i}$ for some $a_1, \ldots, a_n \in \mathbf{Z}$. It follows that the character group $X(T) = \mathrm{Hom}(T, \mathbf{C}^*)$ of a real Lie group T which is isomorphic to the torus T^n is a free abelian group of rank n. Every finite dimensional T-module V can be decomposed as a direct sum of submodules:

$$V = \bigoplus_{\lambda \in X(T)} V_\lambda, \qquad V_\lambda = \{v \in V : \forall t \in T : t \cdot v = \lambda(t)v\},$$

and V is determined up to a T-module isomorphism by specifying $\dim_{\mathbf{C}} V_\lambda$ (the *multiplicity* of λ in V) for all $\lambda \in X(T)$. The *global weights* of V (with respect to T) are defined to be those $\lambda \in X(T)$ for which $V_\lambda \neq \{0\}$. The space V_λ is called the *weight space* in V of weight λ.

Denote the kernel of the exponential map $\exp : \mathcal{L}T \to T$ by I, and put $\mathcal{L}T^* = \mathrm{Hom}_{\mathbf{R}}(\mathcal{L}T, \mathbf{R})$. The *lattice of integral forms* on $\mathcal{L}T$ is defined to be the group

$$\Lambda_0 = \{\lambda \in \mathcal{L}T^* : \lambda(I) \subset \mathbf{Z}\}.$$

There is a natural bijection between $X(T)$ and Λ_0 defined by associating to every $\theta \in X(T)$ the integral form $\alpha \in \Lambda_0$ such that $\theta(\exp x) = e^{2\pi i \alpha(x)}$. The associated integral form of a global weight of a representation of T is called a *real weight* of the representation.

1.3.5 Compact real Lie groups. In the representation theory of compact real Lie groups can be dealt with using a geometrical and analytical approach. The main instrument needed is integration on manifolds. For instance it is easy to prove that every finite dimensional representation V of a compact real Lie group is semisimple, by constructing a G-invariant Hermitian form on V, so that every G-submodule has a G-invariant orthoplement.

Let G be a connected compact real Lie group. A toral subgroup of G is a Lie subgroup of G that is isomorphic to T^n for some n and a maximal torus T of G is a toral subgroup properly included in no other. The *Weyl group* of a maximal torus T is the group $\mathcal{W} = N_G(T)/T$, where $N_G(T) = \{x \in G : xTx^{-1} = T\}$. The Weyl group is finite and it acts faithfully on $\mathcal{L}T$. It can be shown that the maximal torus of G is unique up to conjugation, and that every element of G lies in some maximal torus. In particular all maximal tori have canonically isomorphic Weyl groups, and all maximal tori contain the center of G.

A representation of G on a complex vector space V induces by restriction a representation of the maximal torus T, and thus it gives rise to a set of real weights in $\mathcal{L}T^*$. After constructing a $\mathcal{W}$-invariant inner product on $\mathcal{L}T^*$, it turns out that the non-zero real weights (with respect to T) of the adjoint representation of G form a root system Φ in the euclidean space E generated in $\mathcal{L}T^*$. Furthermore, $\mathcal{W}$ is just the Weyl group of Φ (Bröcker [6], Theorem V(3.12)). The *root lattice* Γ is the lattice in Λ_0 spanned by the roots and its rank is said to be the *semisimple rank* of G. The orthogonal projection in E of the lattice of integral forms Λ_0 lies in the weight lattice lattice Λ of Φ.

1.3.6 Classification of compact real Lie groups. As all maximal tori in a compact real Lie group G are conjugate, the root system Φ is determined by G. As the kernel of the adjoint representation of G is $Z(G)$, the semisimple rank of G is equal to the dimension of T if $Z(G)$ is discrete, or equivalently if G has no abelian connected subgroup other than $\{e\}$. In this case we say that G is semisimple. If G is semisimple, the fundamental group of G is isomorphic to Λ/Λ_0, and its centre $Z(G)$ is isomorphic to Λ_0/Γ. In particular, the lattice of integral forms is just the lattice of abstract weights if G is simply connected and semisimple. Conversely, given a root system Φ with root lattice Γ and weight lattice Λ, and given an intermediary group Λ_0 of $\Gamma \subset \Lambda$, there is a unique compact real semisimple Lie group with root system Φ, and Λ_0 as the lattice of integral forms.

For the general case one can show that every real compact Lie group G is the homomorphic image with finite central kernel of $S \times \hat{G}$ where $S = Z(G)^\circ$ (a torus), and $\hat{G}$ is the universal cover of the commutator subgroup G' of G (which is semisimple). So in this case $\hat{G}$ is uniquely determined by the root system of G, and $S \cong T^{n-r}$ if T has dimension n, and r is the semisimple rank of G.

1.3.7 Representations of compact real Lie groups. Irreducible representations of G have a unique weight whose orthogonal projection in E is dominant (with respect to a fixed maximal torus T and a basis of the root system), called the highest weight of the representation, and conversely every integral form of $\mathcal{L}T$ with this property occurs as the highest weight of an irreducible G-module. The weights of a representation are permuted by $\mathcal{W}$ and multiplicities are preserved. One can now derive Freudenthal's multiplicity formula and Weyl's dimension formula in an analytic manner.

1.3.8 Reductive groups. Let G_c be a complex Lie group and $\gamma : G \to G_c$ a homomorphism of real Lie groups. We say that γ is a *complexification* of G if for every

complex Lie group H and every homomorphism of real Lie groups $f : G \to H$ there is a unique homomorphism of complex Lie groups $f_c : G_c \to H$ such that $f_c \circ \gamma = f$ (the functor "$-_c$" is the left adjoint of the forget functor from complex to real Lie groups).

A complex Lie group G is said to be reductive if G is connected, G has a faithful representation and every finite dimensional representation of G is semisimple. See Hochshild [9] for a proof that the reductive complex Lie groups are just the complexifications of compact real Lie groups. Thus the classification theory and the representation theory can be dealt with as in the compact real case. In fact the complex reductive groups are algebraic groups in the sense of the next section, so one can also use a purely algebraic approach without any reference to the analytic structure.

Finally one can also treat the complex Lie groups using the theory of Lie algebras over $\mathbb{C}$: we say that G is semisimple if $\mathcal{L}G$ is, and define the root system of G to be the root system of $\mathcal{L}G$. See Varadarajan [21] for this approach.

1.4. Linear algebraic groups.

We give a short summary of the structure theory of reductive algebraic groups. The techniques from algebraic geometry that are needed also give useful results on Lie algebras. The theory of algebraic groups gives more results in the prime characteristic case, and more detailed information is obtained in the zero characteristic case than with pure Lie algebra theory. The classification and representation theory again uses root systems and weights. Main references: Humpreys [11], Springer [19] and Waterhouse [23]

1.4.1 Definitions. A *linear algebraic group* (or affine group scheme) over an algebraically closed field K is an affine algebraic variety G over K with a group structure such that the mappings $(x, y) \mapsto xy$ and $x \mapsto x^{-1}$ are morphisms of affine varieties $G \times G \to G$ and $G \to G$. As we will work over algebraically closed fields K only, we will identify an algebraic variety over K with its set of K-valued points (with Zariski topology and a sheaf of regular functions). The *affine algebra* $K[G]$ of G is the K-algebra consisting of the global regular functions on G. The multiplication map $G \times G \to G$ induces a map $K[G] \to K[G] \otimes_K K[G]$ called comultiplication, making $K[G]$ into a so-called *Hopf-algebra* over K, and conversely every Hopf-algebra over K that is reduced and of finite type over K can be obtained in this way.

A *homomorphism* of algebraic groups is a morphism of varieties preserving the group structures. The kernel and image of a homomorphism $G \to G'$ of algebraic groups are closed algebraic subgroups of G and G' respectively. An algebraic group G is said to be *connected* if it is irreducible as a variety. The irreducible component G° of the identity element in G is a normal subgroup of finite index in G, and the connected components of G (in the Zariski topology on G) coincide with the irreducible components: they are the cosets of G°.

1.4.2 Examples. The additive group $\mathbb{G}_a = K$ and the multiplicative group $\mathbb{G}_m = K^*$ are clearly algebraic groups. The n-dimensional *torus* T^n is defined to be $\mathbb{G}_m^n$, and the

n-dimensional *vector group* $\mathbb{A}^n$ is defined to be $\mathbf{G}_a^n$. The *general linear group* $GL(V)$ of a finite dimensional vector space V over K is the multiplicative algebraic group consisting of all K-linear automorphisms of V, and the *special linear group* $SL(V)$ of V is the group $\{x \in GL(V) : \det x = 1\}$. We write $GL(n,K)$ and $SL(n,K)$ for $GL(K^n)$ and $SL(K^n)$.

1.4.3 The Lie algebra of an algebraic group. As is the case with Lie groups, the Lie algebra of an algebraic group G can be obtained as the tangent space in $1 \in G$, or as the space of left invariant derivations of the algebra of global functions. The first point of view implies that the Lie algebra is of the same dimension as G, and the second provides a Lie-bracket operation. First note that G acts on $K[G]$ by letting $(xf)(y) = f(x^{-1}y)$, for $x, y \in G$ and $f \in K[G]$. Now define the Lie algebra $\mathcal{L}G$ as the set of derivations $\delta : K[G] \to K[G]$ satisfying $x\delta(f) = \delta(xf)$ for all $f \in K[G]$, with the operation $[\delta_1, \delta_2] = d_1 \circ \delta_2 - \delta_2 \circ \delta_1$. For example, for $G = \mathbf{G}_a$ the only derivations of $K[G] = K[X]$ (the polynomial ring in one variable) invariant under all translations are of the form $a\frac{\partial}{\partial X}$ with $a \in K$, and for $G = \mathbf{G}_m$ we have $K[G] = K[X, X^{-1}]$, and $\mathcal{L}G = \{aX\frac{\partial}{\partial X} : a \in K\}$.

Recall that the ring O_x of regular local functions at a point x of a variety is a local ring with maximal ideal $\mathfrak{m}_x = \{f \in O_x : f(x) = 0\}$ and residue class field $O_x/\mathfrak{m}_x = K$. As the germ δ_x of a derivation $\delta \in \mathcal{L}G$ in the ring O_1 of local functions at $1 \in G$ is completely determined by its action on $\mathfrak{m}_1$, and $\delta_1(\mathfrak{m}_1^2) = 0$, we obtain a K-linear map from $\mathcal{L}G$ to the tangent space $(\mathfrak{m}_1/\mathfrak{m}_1^2)^*$ at $1 \in G$, which can be proved to be an isomorphism. We know from algebraic geometry that this tangent space has the same dimension as G because all local rings are isomorphic via translations, and the dimensions must agree on an open dense subset of G.

For every homomorphism $f : G \to G'$ of algebraic groups the derivative at $1 \in G$ gives a map $df : \mathcal{L}G \to \mathcal{L}G'$ which is a homomorphism of Lie algebras. For example, the Lie algebra of $GL(n,K)$ is $\mathfrak{gl}(n,K)$, and the derivative of the map $GL(n,K) \to GL(1,K) = K^*$ defined by $x \mapsto \det x$ has derivative $\mathrm{Tr} : \mathfrak{gl}(n,K) \to \mathfrak{gl}(1,K) = K$.

1.4.4 Modules and representations. An *action* of an algebraic group G over K on a variety V is a morphism of varieties $G \times V \to V$ which is a group action of G on V. If H is a closed subgroup of G the quotient space G/H consisting of the left cosets of H in G can be given the structure of a variety in a natural way. In particular we obtain an action of G on G/H, and if H is a normal subgroup, then G/H becomes an algebraic group.

Every finite dimensional vector space V over K has a natural structure of a variety over K, namely that of a vector group. A G-module structure on V is an action of G on V that is K-linear, which is equivalent to giving a representation of G on V, i.e. a homomorphism of algebraic groups $G \to GL(V)$. Again we have the notions of submodules, irreducible modules, semisimple modules, etc. (cf 1.2.1).

For $x \in G$ the derivative of the conjugation map $G \to G$ defined by $y \mapsto x^{-1}yx$ is an automorphism of $\mathcal{L}G$, so we obtain a homomorphism $\mathrm{Ad} : G \to GL(\mathcal{L}G)$, which is readily seen to be algebraic. This is the *adjoint representation* of G. The derivative of any representation $G \to GL(V)$ of G on a finite dimensional vector space V is a

representation $\mathcal{L}G \to \mathfrak{gl}(V)$ of Lie algebras (see 1.2.1). An easy computation shows that $\mathrm{ad} : \mathcal{L}G \to \mathfrak{gl}(\mathcal{L}G)$ is the derivative of the adjoint representation of G.

1.4.5 Jordan decomposition. An endomorphism α is of a vector space V over K is said to be *semisimple* if it is diagonalizable. We say that α is *nilpotent* if $\alpha^n = 0$ for some n, and *unipotent* if $\alpha - \mathrm{id}_V$ is nilpotent. In an algebraic group G every element $x \in G$ can be decomposed uniquely as $x = x_s x_u$ for commuting $x_s \in G$ and $x_u \in G$ such that for every representation $G \to GL(V)$ the element x_s maps to a semisimple endomorphism, and x_u to a unipotent endomorphism. This is called the *Jordan-decomposition* in G. It is preserved under homomorphisms of algebraic groups. For example we have $x = x_s$ in $\mathbf{G_m}$ and $x = x_u$ in $\mathbf{G_a}$.

The Jordan decomposition is used in structure theory of algebraic groups. For instance, the subset G_u of al unipotent elements in a connected algebraic group G is a closed normal subgroup if G is solvable, and the subset G_s consisting of all semisimple elements of G is a closed subgroup if G is nilpotent (that is if G has a chain of normal subgroups with abelian factors). If G is nilpotent (so in particular if G is abelian), then $G = G_u \times G_s$.

1.4.6 The root system. The representation theory for the n-dimensional *torus* T^n is easy: a representation $T^n \to GL(V)$ must map T^n to a commuting set of diagonalizable automorphisms of V, so the image of T^n can be simultaneously diagonalized. It follows that the T^n-module V decomposes as a direct sum of one-dimensional submodules, and that an irreducible representation of T^n is just a homomorphism $T^n \to \mathbf{G_m}$, so it is always of the form $(x_1, \ldots, x_n) \mapsto \prod_{i=1}^n x_i^{a_i}$ for some $a_1, \ldots, a_n \in \mathbf{Z}$. More abstractly, a representation V of a torus T decomposes as

$$V = \bigoplus_{\lambda \in X(T)} V_\lambda, \qquad V_\lambda = \{v \in V : \forall t \in T : t \cdot v = \lambda(t)v\},$$

where $X(T)$ is the group $\mathrm{Hom}(T, \mathbf{G_m})$. The $\lambda \in X(T)$ for which $V_\lambda \neq \{0\}$ are the *weights* of the representation, and $\dim_K V_\lambda$ is the *multiplicity* of λ in V.

Let G be an algebraic group. A toral subgroup of G is a closed subgroup of G that is isomorphic to T^r for some r and a maximal torus T of G is a toral subgroup properly included in no other. The *Weyl group* of a maximal torus T is the group $\mathcal{W} = N_G(T)/C_G(T)$, where

$$N_G(T) = \{x \in G : xTx^{-1} = T\}, \qquad C_G(T) = \{x \in G : \forall t \in T\ xtx^{-1} = t\}.$$

It can be shown that the maximal torus of G is unique up to conjugation, and that every semisimple element of G lies in some maximal torus. In particular all maximal tori have canonically isomorphic Weyl groups.

A representation of G on a vector space V induces by restriction a representation of the maximal torus T, and thus it gives rise to a set of weights in the *weight lattice* $X(T)$. The *roots* of G (with respect to T) are defined to be the non-zero weights of the adjoint representation. The root lattice Λ_r is the lattice spanned by the roots and its

rank is said to be the *semisimple rank* of G. We make the real vector space $\mathbf{R} \otimes_{\mathbf{Z}} X(T)$ into a euclidean space by constructing a $\mathcal{W}$-invariant inner product, and this way the roots become a root system in $E = \mathbf{R} \otimes_{\mathbf{Z}} \Lambda_r$ and the orthogonal projection of the weight lattice $X(T)$ in E lies in the lattice Λ of abstract weights. The action of the the Weyl group of T on $X(T)$ gives rise to a canonical isomorphism between the Weyl group of T and the Weyl group of the root system of T.

1.4.7 Classification of reductive groups. We define the *radical* $R(G)$ of an algebraic group G to be the maximal solvable normal connected subgroup of G, and we say that G is *semisimple* if $R(G) = \{1\}$. If G is solvable as a group, then $G_u = \{x_u : x \in G\}$ is a closed subgroup, so the unipotent radical $R_u(G) = R(G)_u$ is a closed subgroup of G. A *reductive group* is an linear algebraic group G with $R_u(G) = \{1\}$.

It is not hard to show that the root system (and the weight lattice) of an algebraic group G and of $G/R_u(G)$ coincide, we can obtain no information about $R_u(G)$ from the root system. However, we have the following uniqueness theorem: Let G and G' be reductive algebraic groups with maximal tori T and T', and suppose we have an isomorphism $X(T) \to X(T')$ of abelian groups such that the root system of G is mapped isomorphically to that of G'. Then there is an isomorphism $G' \to G$ whose restriction to T is an isomorphism $T' \to T$ inducing this map $X(T) \to X(T')$.

Finally, every reductive group is the homomorphic image with finite central kernel of $T^n \times G_1 \times \cdots \times G_m \to G$, where the G_i are semisimple algebraic groups whose root systems are irreducible and whose weight lattices coincide with the abstract weight lattices. For example, the natural map $T^1 \times SL(n, K) \to GL(n, K)$ has a cyclic kernel of order n. Furthermore, we know that the centre of each G_i is isomorphic (as a group) to the fundamental group Λ/Λ_r of its root system.

1.4.8 Representations of reductive groups. Just as with Lie algebras, the weights of a representation of a reductive group G are permuted by the Weyl group and the multiplicities are preserved. Irreducible representations of G have a unique weight whose orthogonal projection in E (the real subspace of $X(T) \otimes_{\mathbf{Z}} \mathbf{R}$ generated by the roots) is dominant, called the highest weight of the representation. For the case that $\mathrm{char} K = 0$ one can use the theory of Lie algebras to show that all representations are semisimple and one can use Freudenthal's multiplicity formula and Weyl's dimension formula to obtain more precise information about dimensions and multiplicities.

In characteristic $p > 0$ it is still true that irreducible modules are given by highest weights, but not all finite-dimensional representations are semisimple, and far less is known about the dimensions of irreducible modules given by a highest weight. All formulas and algorithms given in the sequel will hold generally for the characteristic zero case, but only for a few situations in positive characteristic the outcome will be the same. For instance the highest weight of an irreducible module always occurs with multiplicity 1, so Weyl's dimension formula remains valid for irreducible modules corresponding to *minimal weights*, as all weights of such a module lie in the Weyl group orbit of the highest weight.

1.4.9 Complex reductive groups. In this final section the relation between algebraic

groups over $\mathbb{C}$ and reductive complex Lie groups is indicated as in Serre [18]. An algebraic group over $\mathbb{C}$ always has a canonical structure of a complex Lie group (any smooth algebraic variety over $\mathbb{C}$ has an analytic structure). Conversely, every reductive complex Lie group G has a unique structure of an algebraic group over $\mathbb{C}$ that is compatible with its algebraic structure, and every homomorphism from G to the Lie group associated with an algebraic group over $\mathbb{C}$ is algebraic. This algebraic structure can be obtained by fixing a faithful representation $G \to GL(n, \mathbb{C})$ and verifying that the image is an algebraic subgroup. The algebraic structure on G can also be defined in a more intrinsic manner, by letting the Hopf-algebra of G be the algebra of holomorphic functions on G whose translates generate a finite dimensional vector space over $\mathbb{C}$, with comultiplication defined by the multiplication map $G \times G \to G$.

Chapter 2. The LiE package

In this chapter an overview is given of the LiE package and in particular of those features of LiE that enable the user to do calculations with representations in terms of root systems and weights. Many examples are given of how LiE can be used. When starting up LiE the following message is printed:

```
LiE      version 1.1 created at Thu May 18 14:16:07 MET DST 1989
Authors: Arjeh M. Cohen, Bert Lisser, Bart de Smit, Ron Sommeling

type '?' for information
type '?functions' for a list of functions.
>
```

All commands for LiE are given on a line following the LiE -prompt ">", so in all examples, the lines starting with ">" are entered by the user.

It is not our intention to give a rigorous description of the LiE language, and the use of variables and "for"-loops in the examples are supposed to be self-explanatory. See the LiE manual [13] for more details on syntax and the programming language. The main objective here is to describe the kind of calculations that the user of LiE can perform. In the next chapter some of the implemented algorithms are explained.

2.1. Objects in LiE

The main objects of computation in LiE are integers, vectors, matrices, and groups. In this section the use and notation of each is briefly explained.

2.1.1 Integer. *Input notation*: a string of digits possibly headed by a minus sign. An integer can have arbitrary length.

Operations: +, -, * (multiplication), ^ (exponentiation), / (integral division: rounded of downwards), and % (remainder of integral division: "mod"-operator). A simple factorisation procedure is included for convenience. Only prime factors smaller than 2^{15} are found. Example:

```
> a=3^50 - 2^50
> a
717897986565952681927625
> factor(a)
5^3 * 11 * 101 * 211 * 1201 * 20399203039801
(Last factor need not be a prime)
>
```

2.1.2 Vector. All vector entries are integers, but in contrast with the "integer object", the entries do not have arbitrary length. This option was chosen because of speed

and memory storage reasons, bearing in mind that most of the computations we are concerned with involve matrices and vectors with relatively small integers. Since there is no overflow check in most of the algorithms involved, the user should always be aware of possible overflow.

Input notation: `[` followed by a row of integers, which are separated by commas, and ending with `]`.

Operations: If `v` is a vector and `i` an integer, then the number `v[i]` is the `i`-th entry of `v`, and `v-i` is the vector that is obtained from `v` by leaving out the `i`-th entry. The length of a vector `v` is available as `size(v)`. Two vectors can be pasted by the operator `^`.

Example:

```
> v=[21,-3,0,21,1]
> v[2]
    -3
> size(v)
     5
> v-2
[ 21 0 21 1 ]
```

```
> v^[1,2,3]
[ 21 -3 0 21 1 1 2 3 ]
> sort(v)
[ 21 21 1 0 -3 ]
> -sort(-v)
[ -3 0 1 21 21 ]
>
```

2.1.3 Matrix. All matrix entries are integers with the same restriction on the length of these integers as with vector entries.

Input notation: `[` followed by a row of vectors which are separated by commas, and ending with `]`.

Operations: If `a` is a matrix and `i` and `j` are integers, then the vector `a[i]` is the `i`-th row, and `a[i,j]` is the `(i,j)`-th element of the matrix `a`. The number of rows of a matrix `m` is available as `rowsize(m)`. The operations mentioned for vectors all have the obvious meaning for matrices, when one thinks of a matrix as a "vector of vectors".

Beware of overflow! Example:

```
> m=[[1,1],[0,1]]
> m
| 1 1 |
| 0 1 |

> m[1]

[ 1 1 ]
> m[2]
[ 0 1 ]
> m[2,1]
     0
```

```
> m^300
| 1 300 |
| 0   1 |

> m^10000000000
| 1 1410065408 |
| 0          1 |

>
```

2.1.4 Group. As mentioned in the previous chapter, an isomorphism class of reductive complex Lie algebras is described by a type of root system and the dimension of its centre. Furthermore, a compact real Lie group or a reductive complex Lie group that is the product of a torus and a simply connected semisimple Lie group is characterized by the same information. Using the notation of the classification of root systems, a *group* in LiE is defined to be a string of upper case letters followed by numbers subject to the following conditions. The letters are from {A, B, C, D, E, F, G, T}, and each letter is followed by a positive integer, with the conditions as in the classification of irreducible root systems (for instance, E can only be followed by 6, 7 or 8). The letter T stands for torus and the number following it for its dimension. In terms of Lie groups, the full string stands for the direct product of the simply connected simple Lie groups and tori of which it is built up. Example: A3T4B12A4T6A3E7 stands for the complex Lie group

$$SL(4,\mathbf{C}) \times (\mathbf{C}^*)^4 \times Spin(25,\mathbf{C}) \times SL(5,\mathbf{C}) \times (\mathbf{C}^*)^6 \times SL(4,\mathbf{C}) \times E_7(\mathbf{C}).$$

Note: The actual group that LiE will work with is A3B12A4A3E7T10. In other words, the order of the factors does not change but for those containing part of the central torus. The full central torus is collected and listed at the end.

If g is a group then g[0] is the toral component, and for every positive integer i less than or equal to the number of simple components of g, the i-th component of g is available as g[i]. The function diagram returns the Dynkin diagram of a group, with an enumeration of the simple roots. Example:

```
> g=T3B6T5E8
> g[0]
     T8
> g
     B6E8T8
> g[2]
     E8
> diagram(g)
0---0---0---0---0---0=>=0
1   2   3   4   5   6
B6

          0 8
          |
          |
0---0---0---0---0---0---0
7   9   10  11  12  13  14
E8

8-dim central torus
>
```

2.2. Root systems

In this first section on the built in functions of LiE the main procedures for calculations with root systems are introduced. Recall that a group in the sense of the previous section determines a root system. For each root system a basis of simple roots is fixed (this choice is unique up to an automorphism of the root system, i.e. an element of the Weyl group).

2.2.1 Inner product. The inner product of the underlying euclidean space of a root system Φ defined by a group g is determined uniquely up to multiplication by a scalar. The function `inprod(v1,v2,g)` returns the normalized inner product of two elements `v1` and `v2` in the root lattice Λ represented as coordinate vectors with respect to the basis $\alpha_1, \ldots, \alpha_r$ of simple roots, the normalisation being that of Bourbaki (see [1]). The function `norm(v,g)` is just `inprod(v,v,g)`.

The expression $\langle x, \alpha \rangle = 2(x,\alpha)/(\alpha,\alpha)$ with $x \in \Lambda$ and $\alpha \in \Phi$ is always integral, and it does not depend upon the chosen normalisation of the inner product. The Cartan matrix $(\langle \alpha_i, \alpha_j \rangle)_{i,j}$ can be calculated by the function `cartan(g)`. The same information can be inferred from the Dynkin diagram, which is printed by the procedure `diagram(g)`. Example:

```
> m=id(2)
> m
| 1 0 |
| 0 1 |

> for i=1 to 2 do print(norm(m[i],G2)) od
    2
    6
> cartan(G2)
|  2 -1 |
| -3  2 |

> for i=1 to 2 do for j=1 to 2 do print(inprod(m[i],m[j],G2)) od od
    2
   -3
   -3
    6
>
```

The Cartan matrix expresses the simple roots in terms of the fundamental weights λ_i as the λ_i are defined by $\langle \lambda_i, \alpha_j \rangle = \delta_{i,j}$. Thus the index of the root lattice in the weight lattice is the determinant of the Cartan matrix, and it is available in LiE as `detcartan(g)`. Finally, `icartan(g)` returns the inverse matrix of the Cartan matrix, multiplied by `detcartan(g)` in order to clear the denominators of its coefficients.

2.2.2 Positive roots. Once a basis of simple roots is fixed, all roots in a root system can be written as a $\mathbb{Z}$-linear sum of simple roots with either all coefficients non-negative (in which case the root is said to be positive) or all non-positive. The function `posroots(g)` returns a matrix whose rows are the positive roots of the root system of g expressed in coordinates with respect to the basis of simple roots. Consequently, the call `posroots(g)*cartan(g)` yields the positive roots with respect to the basis of fundamental weights.

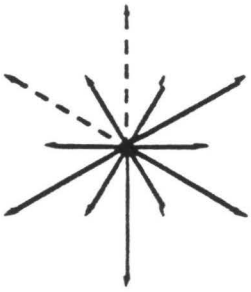
The functions `lierank(g)` and `numproots(g)` return the dimension of the underlying euclidean space, and the number of positive roots, respectively. The function `highroot(g)` returns the unique largest root with respect to the partial order $\prec$ introduced in 1.1.1. Example:

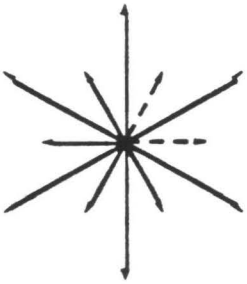
```
> posroots(A2G2)
| 1 0 0 0 |
| 0 1 0 0 |
| 1 1 0 0 |
| 0 0 1 0 |
| 0 0 0 1 |
| 0 0 1 1 |
| 0 0 2 1 |
| 0 0 3 1 |
| 0 0 3 2 |

> highroot(G2)
[ 3 2 ]
> numproots(G2)
     6
>
```

2.2.3 Subsystems. A subsystem Ψ of a root system Φ is a subset of roots that is a root system itself with respect to the inner product. We say that the subsystem is *closed* if for all $x, y \in \Psi$ with $x + y \in \Phi$ we have $x + y \in \Psi$. For example, in the root system of type G_2 both the set of long roots and the set of short roots form a subsystem of type A_2, but only the set of long roots forms a closed subsystem.

Let a set of roots in the root system specified by a group g be given as a matrix r whose rows are interpreted as coordinate vectors with respect to the simple roots of g. The procedure `fundam(r,g)` then computes a base of the subsystem generated by these roots (i.e. of the smallest subsystem containing them), and `closure(r,g)` finds a base of the generated closed subsystem. The group that is determined by the subsystem whose simple roots are given as rowvectors of a matrix m can be recovered from m by the call `carttype(m,g)`. In particular, all positive roots in the subsystem generated by a given set of roots can be obtained by the call `posroots(carttype(fundam(r,g)))*fundam(r,g)` and replacing `fundam` by `closure` we get the positive roots of the closed subsystem generated by r.

As an example we look at the subsystems of the root system of G_2 given by long and short roots respectively.





```
> m=[[3,2],[0,1]]
> fundam(m,G2)
| 3 1 |
| 0 1 |

> c=closure(m,G2)
> c
| 3 1 |
| 0 1 |

> carttype(c,G2)
    A2
```

```
> n=[[2,1],[1,0]]
> fundam(n,G2)
| 1 1 |
| 1 0 |

> c=closure(n,G2)
> c
| 1 0 |
| 0 1 |

> carttype(c,G2)
    G2
```

2.3. Weyl group

The elements of the Weyl group of a root system can be represented in several ways. First of all, an element of the Weyl group is determined by its action on the root lattice or the weight lattice, so we can represent it as a matrix of this linear action with respect to the simple roots or the fundamental weights. Alternatively, such an element can always be written as a composition of a number of reflections in simple roots, called a Weyl word. In the LiE package the action of the Weyl group on roots and weights is always a right action, as they are represented by row vectors, and matrices operate from the right on row vectors. Denoting the reflection in the i-th simple root α_i by r_i, this means that the action of the Weyl group element $r_1 r_2$ is to first reflect in the first simple root, and then in the second. The presentation of the Weyl group in terms of the reflections r_i is:

$$\mathcal{W} = \langle r_1, \ldots, r_m \mid \forall i,j \in \{1,\ldots,n\} : (r_i r_j)^{m_{i,j}} = 1 \rangle,$$

where the $m_{i,j} \in \mathbb{Z}_{>0}$ are define by

$$(\alpha_i, \alpha_j) = \sqrt{(\alpha_i,\alpha_i)(\alpha_j,\alpha_j)}\cos(2\pi/m_{i,j}).$$

The order of the Weyl group of the root system specified by a group g is available as `worder(g)`. Example:

```
> w=worder(E8D12)
> w
6834884468067532800000
> factor(w)
2^35 * 3^10 * 5^4 * 7^2 * 11
>
```

2.3.1 Weyl words. A weyl word for a group g is a vector of arbitrary length whose entries are positive integers not exceeding `lierank(g)`. Every Weyl word for g represents an element in the Weyl group; for instance `[1,2]` stands for $r_1 r_2$ if the rank of g is at least two. A Weyl word is said to be reduced if there is no Weyl word of shorter length that represents the same element of the Weyl group. Not every element in the Weyl group has a unique reduced Weyl word representing it, so there are different strategies possible to obtain a reduced Weyl word from a given Weyl word. One such strategy is implemented in **LiE** : the procedure `reduce(v,g)`. The length `size(reduce(v,g))` of the reduced Weyl word is also available as `length(v,g)`. The function `longword(g)` returns the longest reduced Weyl word of group g. Example:

```
> reduce([1,2,3,2,2,3,2],D3)
[ 1 ]
> length([1,2,3,1,2,3,1,2,3,3,2,1],D3)
    6
> reduce([1,2,3,1,2,3,1,2,3,3,2,1],D3)
[ 2 3 1 3 2 1 ]
> longword(D3)
[ 1 2 1 3 1 2 ]
>
```

2.3.2 Action on root lattice and weight lattice. The action on the root lattice of a Weyl group element given by a Weyl word v is given by the matrix `weylmat(v,g)` on the simple roots. Conversely, the function `wword(m,g)` returns a Weyl word whose action is given by the matrix m with respect to the basis of fundamental weights, so the call

```
wword((icartan (g)*weylmat(v)*cartan(g))/detcartan(g))
```

should always return a reduced word representing the same Weyl group element as v. The action of v on a given element r of the root lattice (so r is a coordinate vector with respect to the simple roots) is given by `wrtaction(r,v,g)`, and on an element w of the weight lattice by `waction(w,v,g)`. Obviously, `wrtaction` could be computed from `waction` using `cartan`, but they are both included in **LiE** for convenience of the user. For roots expressed as a linear combination v of simple roots, the function `reflection(v,g)` returns the matrix of the reflection in the hyperplane perpendicular to this root, with respect to the basis of simple roots. Example:

```
> weylmat([1,2],G2)
| -1 -1 |
|  3  2 |

> reflection([1,0],G2)*reflection([0,1],G2)
| -1 -1 |
|  3  2 |

> wrtaction([1,3],[1,2],G2)
[ 8 5 ]
> icartan(G2)*weylmat([1,2],G2)*cartan(G2)
| 2 -1 |
| 3 -1 |

> waction([1,1],[1,2],G2)
[ 5 -2 ]
> wword([[2,-1],[3,-1]],G2)
[ 1 2 ]
>
```

In the current version of LiE (version 1.1), the answer of `wword` is not checked with `weylmat`, so it is possible that `wword(m)` returns a Weyl word even if the matrix `m` does not lie in the Weyl group.

One of the main functions in LiE which is used constantly in calculations with representations (see the next section) is the function `worbit(v,g)` which generates a matrix whose rows form the entire orbit of the weight vector `v` under the action of the weylgroup. The function `wrtorbit` is also available; it computes the orbit of a root vector. If one is only interested in the size of the orbit, the function `worbitsize(v,g)` should be called, as it is much faster than `rowsize(worbit(v,g))`. Example:

```
> worbit([1,1],G2)
|  4 -3 |
| -4  1 |
|  5 -3 |
| -5  2 |
|  1  1 |
| -1  2 |
|  1 -2 |
| -1 -1 |
|  5 -2 |
| -5  3 |
|  4 -1 |
| -4  3 |
```

```
> worbitsize([1,1],G2)
    12
> wrtorbit([1,1],G2)
| -1 -1 |
| -2 -1 |
|  1  0 |
| -1  0 |
|  2  1 |
|  1  1 |

>
```

2.3.3 Dominant weights. In the orbit of every weight there is a unique dominant weight. This dominant representative of the orbit of a weight vector `w` is available in LiE as `dominant(w,g)`. The routine `wword(w,g)` returns a Weyl group element that sends `w` to a dominant weight. Example:

```
> dominant([1,-2],G2)
[ 1 1 ]
> wword([1,-2],G2)
[ 2 1 2 1 2 ]
> waction([1,-2],[2,1,2,1,2],G2)
[ 1 1 ]
>
```

2.4. Representations

As indicated in chapter 1, a representation of a semisimple Lie algebra, compact real Lie group, reductive complex Lie group or reductive algebraic group decomposes as a direct sum of irreducible representations, and these are in turn given by highest weights. In LiE an irreducible representation is represented by its highest weight, expressed as a coordinate vector with respect to the basis of fundamental weights. For instance, the function `adjoint(g)` returns the highest weight of the adjoint representation of a simple group `g`.

Not necessarily irreducible representations can be given by a matrix whose rows consist of a dominant weight and the multiplicity with which its associated highest weight module occurs. Such a matrix will be called a *decomposition matrix*.

2.4.1 The weight system of an irreducible representation. A highest weight module V (i.e. irreducible representation) decomposes as a direct sum of weight spaces V_λ, where λ ranges over all weights in the weight lattice. The λ with $V_\lambda \neq \{0\}$ constitute the weight system of V.

The set of dominant weights of a highest weight module over a group `g` given by a weight `w` can be obtained in LiE as the rows of the matrix `domweights(w,g)`. See 3.3.1 for some remarks on the algorithm. The entire weight system could be obtained by applying `worbit` to each of the rows, as every Weyl group orbit contains exactly one dominant weight, and the weight system is closed under the Weyl group action.

2.4.2 Multiplicities. The multiplicity of a weight λ in a module V is defined as $\dim V_\lambda$. As $V_\lambda \cong V_{\sigma(\lambda)}$ for $\sigma \in \mathcal{W}$, the Weyl group acts on the weight system of V preserving multiplicities, so if one wants to determine all multiplicities of the weights of V it suffices to generate the multiplicities of the dominant weights. This is done by the call `mul(w,g)` in LiE , which gives a list of all dominant weights and their multiplicities of the irreducible module over the group `g` whose highest weight is given by the vector `w`. The call returns a matrix whose rows consist of a dominant weight followed by its multiplicity. If one is only interested in the multiplicity of the weight

v, the call `mul(w,v,g)` should be invoked, which returns the multiplicity as an integer. The algorithm used in `mul` is a slight adaptation of Freudenthal's recursion formula; see the next chapter for more details.

This is one of the points in LiE where the original decision to limit the length of integer entries in vectors and matrices to the standard length of an integer in C might cause overflow. Some attempt has been made to recognize overflow, and LiE then gives a multiplicity of -1. This will be improved in a future version of LiE . Example:

```
> mul([0,0,0,1,0,0,0,1],E8)
| 0 0 0 1 0 0 0 1        1 |
| 1 0 0 0 0 1 0 1        4 |
| 0 1 1 0 0 0 0 0        5 |
| 1 0 0 0 1 0 0 0       18 |
| 0 1 0 0 0 0 1 1       15 |
| 0 1 0 0 0 1 0 0       60 |
| 2 0 0 0 0 0 0 2       15 |
| 0 0 1 0 0 0 0 2       50 |
| 2 0 0 0 0 0 1 0       55 |
| 0 0 1 0 0 0 1 0      175 |
| 1 1 0 0 0 0 0 1      480 |
| 0 0 0 0 0 0 2 1       45 |
| 0 0 0 0 0 1 0 2      145 |
| 0 0 0 0 0 1 1 0      470 |
| 0 0 0 0 1 0 0 1     1230 |
| 3 0 0 0 0 0 0 0      154 |
| 1 0 1 0 0 0 0 0     1239 |
| 0 2 0 0 0 0 0 0     1260 |
| 0 0 0 1 0 0 0 0     3059 |
| 1 0 0 0 0 0 0 3      390 |
| 1 0 0 0 0 0 1 1     2980 |
| 1 0 0 0 0 1 0 0     7145 |
| 0 1 0 0 0 0 0 2     6930 |
| 0 1 0 0 0 0 1 0    16110 |
| 2 0 0 0 0 0 0 1    16098 |
| 0 0 1 0 0 0 0 1    35188 |
| 0 0 0 0 0 0 0 4      980 |
| 0 0 0 0 0 0 1 2    15470 |
| 0 0 0 0 0 0 2 0    35045 |
| 1 1 0 0 0 0 0 0    74872 |
| 0 0 0 0 0 1 0 1    74490 |
| 0 0 0 0 1 0 0 0   154810 |
| 1 0 0 0 0 0 0 2   153627 |
| 1 0 0 0 0 0 1 0   312791 |
| 0 1 0 0 0 0 0 1   619010 |
| 2 0 0 0 0 0 0 0   619598 |
| 0 0 1 0 0 0 0 0  1202887 |
| 0 0 0 0 0 0 0 3   309211 |
| 0 0 0 0 0 0 1 1  1200874 |
| 0 0 0 0 0 1 0 0  2294793 |
| 1 0 0 0 0 0 0 1  4308133 |
| 0 1 0 0 0 0 0 0  7968800 |
| 0 0 0 0 0 0 0 2  7965405 |
| 0 0 0 0 0 0 1 0 14533242 |
| 1 0 0 0 0 0 0 0       -1 |
| 0 0 0 0 0 0 0 1       -1 |
| 0 0 0 0 0 0 0 0       -1 |

>
```

2.4.3 Dimension of a representation. The dimension of a highest weight module V of a group g given by a weight vector w is available in LiE as `deg(w,g)`. It should be equal to the sum

$$\sum_{v} \texttt{mul}(w,v,g) * \texttt{worbitsize}(v,g)$$

where the sum ranges over all dominant weights v of V and in fact this formula has been used to test the functions `mul` and `deg`, as they have been programmed independently (Freudenthal's recursion formula and Weyl's dimension formula, respectively). There is no trouble with overflow in `deg`:

```
> deg([0,0,0,1,0,0,0,1],E8)
919045960000
> deg([1,1,1,1,1,1,1,1],E8)
1329227995784915872903807060280344576
> deg([1,2,3,4,5,6,7],A2G2T3)
137655
>
```

2.4.4 Reducible representations. There are two ways to represent a not necessarily irreducible representation of a group g: as a decomposition matrix or as a multiplicity matrix. In this section two functions are defined in the LiE language that can calculate the dimension of the represented module, and it is shown how the user can obtain a decomposition matrix from a multiplicity matrix. This procedure is essential for the calculation of branch rules.

A decomposition matrix gives the decomposition of a module V in irreducible submodules as follows: the rows of the decomposition matrix are dominant weights followed by an integer denoting the multiplicity in V of the highest weight module corresponding to this weight. In order to obtain the dimension of a representation given by a decomposition matrix, one computes the dimension of every occurring irreducible representation, and adds these up with the correct multiplicities. Thus the dimension of a module over g with decomposition matrix `m` is given by the function `decmatdeg(m,g)` defined in LiE as follows:

```
> decmatdeg(mat m; grp g)= \
    d=0; r=lierank(g); \
    for i=1 to rowsize(m) do \
      d=d+m[i][r+1]*deg(m[i]-(r+1),g) od; \
    d
```

(This is a function definition in the LiE language; the character "\" is necessary to break the line. The operator "-" in "`m[i]-(r+1)`" deletes the last entry of the vector `m[i]`.)

A multiplicity matrix of a representation is a matrix listing the dominant part of the weight system with multiplicities in the usual format: the rows consist of a dominant weight and an integer denoting the multiplicity of the weight in the representation. Consequently the dimension of a module specified by a multiplicity matrix is given by the function

```
> mulmatdeg(mat m; grp g)=\
    d=0; r=lierank(g); \
    for i=1 to rowsize(m) do \
      d=d+m[i][r+1]*worbitsize(m[i]-(r+1),g) od; \
    d
```

In order to obtain the multiplicity matrix of a module whose decomposition matrix is given, one should apply `mul` to each of the rows. Conversely, the procedure `decomp(m,g)` yields the decomposition matrix of the module with multiplicity matrix

m, or it gives the error message "not a module.." if the supposed multiplicity matrix cannot be the multiplicity matrix of a module over g. A less restrictive version of the algorithm used in decomp is vdecomp, which yields a so-called virtual module, i.e. a decomposition matrix with possibly negative multiplicities. Note that for a purely toral group the notions of decomposition and multiplicity matrix coincide.

It is often convenient to have a canonical form for decomposition matrices and multiplicity matrices; for instance when one wants to know whether two decomposition matrices represent isomorphic modules. Such a canonical form can be calculated with the call redmulmat(m) (abbreviation of reduce - multiplicity - matrix): it returns the matrix representing the same multiplicity matrix as m, but sorted according to the lexicographical ordering of rows on all but the last component, and having at most one row with specified entries in all but the last component.

We finish this section with an example. The operation a^b merely puts the rows of the matrices a and b together in one matrix.

```
> a=mul([1,2],G2)
> a
| 1 2   1 |
| 4 0   1 |
| 2 1   2 |
| 0 2   2 |
| 3 0   3 |
| 1 1   5 |
| 2 0   7 |
| 0 1   7 |
| 1 0  10 |
| 0 0  10 |

> deg([1,2],G2)
  86
> b=mul([1,1],G2)
> deg([1,1],G2)
  66
> for i=1 to 5 do \
    b[i,3]=100*b[i,3] od
> b
| 1 1 100 |
| 2 0 200 |
| 0 1 200 |
| 1 0 400 |
| 0 0 400 |
```

```
> tm=redmulmat(a^b)
> tm
| 4 0   1 |
| 3 0   3 |
| 2 1   2 |
| 2 0 207 |
| 1 2   1 |
| 1 1 105 |
| 1 0 410 |
| 0 2   2 |
| 0 1 207 |
| 0 0 410 |

> td=decomp(tm,G2)
> td
| 1 2   1 |
| 1 1 100 |

> mulmatdeg(tm,G2)
  6686
> decmatdeg(td,G2)
  6686
>
```

2.5. Tensor products and branch rules

A large part of LiE was written to enable the user to manipulate with representations. For instance the decomposition matrix of the *direct sum* of two representations given as decomposition matrices `m1` and `m2` can be obtained by the call `addmul(m1,m2)`, which just pastes the two matrices together and then finds the canonical form with `redmulmat` (so it is the same as `redmulmat(m1^m2)`). The computation of tensor products and branching rules is more involved however. The main procedures in LiE for handling tensor products and branch rules are introduced in this section.

2.5.1 Tensor product. There are several calls in LiE calculating tensor product decompositions. First of all, the decomposition matrix of the tensor product of two modules over a simple group `g` given by either two highest weights (if the modules are irreducible) or by two decomposition matrices, is available in LiE as `tensor(m1,m2,g)`. Secondly, the call `tensor(m1,m2,v,g)` returns the multiplicity of the weight `v` in the tensor product. The algorithm used is Klimyk's formula (see section 3.3). Finally the procedure `ptensor(n,m,g)` will return the decomposition matrix of the `n`-th tensor power of the `g`-module specified by `m`, where `m` should either be a highest weight or a decomposition matrix. Example:

```
> tensor([1,0],[0,1],G2)
| 1 1 1 |
| 2 0 1 |
| 1 0 1 |

> p=ptensor(3,[1,0],A2)
> p
| 3 0 1 |
| 1 1 2 |
| 0 0 1 |
```

```
> tensor([[1,0,1]],p,A2)
| 4 0 1 |
| 2 1 3 |
| 1 0 3 |
| 0 2 2 |

> ptensor(4,[1,0],A2)
| 4 0 1 |
| 2 1 3 |
| 1 0 3 |
| 0 2 2 |
```

2.5.2 Symmetric and alternating powers. Symmetric and alternating tensor powers of a highest weight module over a simple group `g` specified by a vector `w` can be computed with `symtensor(n,w,g)` and `alttensor(n,w,g)`, where `n` is an integer indicating the tensor power. The calculation is done recursively by the formula

$$\texttt{symtensor(n,w,g)}^{\texttt{n}} = \bigoplus_{\texttt{k}} \texttt{symtensor(n-k,w,g)} \otimes \texttt{adams(k,w,g)},$$

where the direct sum $\bigoplus$ ranges over all `k` with $1 \leq \texttt{k} \leq \texttt{n}$. Here the Adams operation `adams(k,w,g)` returns the multiplicity matrix of the virtual module whose dominant weight decomposition matrix is obtained from the dominant weight multiplicity matrix

of w by multiplication by k of the weights (i.e., all entries but those in the last column). Changing the sign (i.e. the sign of the multiplicities) of the k-th summand for even k, we obtain the recursion formula for alternating powers. Example:

```
> symtensor(2,[1,1],A2)
| 2 2 1 |
| 1 1 1 |
| 0 0 1 |

> alttensor(2,[1,1],A2)
| 3 0 1 |
| 1 1 1 |
| 0 3 1 |
```

```
> ptensor(2,[1,1],A2)
| 2 2 1 |
| 3 0 1 |
| 0 3 1 |
| 1 1 2 |
| 0 0 1 |

>
```

$G_1 \subset G_2 \Rightarrow \Lambda_2 \subset \Lambda_1$

2.5.3 Branching. Given a subgroup G_1 of a group G_2, every G_2-module V is by restriction of scalars a G_1-module (this process is called *branching*). If we choose the maximal torus T_1 of G_1 in such a way that it contains a maximal torus T_2 of G_2, then we obtain by restriction a map $r : \Lambda_2 \to \Lambda_1$ of the weight lattice Λ_2 of G_2 to the weight lattice Λ_1 of G_1. Once this map is given by a matrix with respect to the bases of fundamental weights the procedure branch in LiE can calculate the decomposition of a highest weight module over G_2 into irreducible G_1-modules. The call has 4 parameters: branch (w,g1,m,g2) returns the decomposition matrix of the g1-module that is obtained from the irreducible g2-module with highest weight w, by the restriction of the weight lattice of g2 to that of g1 given by the matrix m. The present implementation insists that g2 is a simple group. Warning: LiE does not check whether the given restriction matrix is indeed the matrix of restriction to a subgroup, so if this is not the case it may give answers that do not make sense or it may give curious error messages. Example:

```
> branch([1,1],T2,id(2),A2)
|  0  0 2 |
| -1 -1 1 |
|  1 -2 1 |
| -2  1 1 |
|  2 -1 1 |
| -1  2 1 |
|  1  1 1 |
```

```
> m=[[1,1],[0,1]]
> b=branch([2,1],A2,m,G2)
> b
| 3 2 1 |
| 2 3 1 |
| 2 1 1 |
| 1 2 1 |
| 0 1 1 |
| 1 0 1 |
```

The form in which the subgroup needs to be specified in order to use the branch-call is not always readily available. For instance, if one identifies a subgroup by looking at a sub-diagram of the (extended) Dynkin diagram, it is not always trivial to obtain

the correct restriction matrix. The two ways offered by LiE to obtain this matrix are treated next.

2.5.4 Branching to fundamental subgroups. A matrix `m` whose rows are the simple roots of a closed subsystem of the root system of a simple group `g` determines a subgroup of type `carttype(m,g)` of `g`. The subgroups obtained in this way are said to be the fundamental subgroups of `g`. See Borel & De Siebenthal [2] for a proof that all subgroups of maximal rank are fundamental subgroups, and that the subsystem can be constructed by leaving out a vertex in the extended Dynkin diagram. Recall that such a matrix `m` can be obtained from the procedure `closure`, which computes the simple roots of the smallest closed subsystem that contains a given set of roots (see 2.2.3). Given the matrix `m`, the restriction matrix needed for the branch call is available as `resmat(m,g)`. For instance, one can obtain the restriction matrix of the subgroup A_2 of G_2 given by the long roots as follows:

```
> c=closure([[0,1],[3,2]],G2)
> c
| 3 1 |
| 0 1 |

> carttype(c,G2)
    A2
> m=resmat(c,G2)
> m
| 1 0 |
| 1 1 |
```

```
> branch([0,3],A2,m,G2)
| 3 3 1 |
| 2 3 1 |
| 3 2 1 |
| 2 2 1 |
| 1 3 1 |
| 3 1 1 |
| 1 2 1 |
| 2 1 1 |
| 0 3 1 |

>
```

2.5.5 The `MAXSUB`-file. The file "`MAXSUB`" that comes with the LiE package is a library containing for each simple group G of rank less than or equal to 8 a list of maximal subgroups of smaller rank than G. This list was copied from [1], pp. 260-262, and it is believed to be complete... In particular, it should contain all maximal non-fundamental subgroups for simple groups up to rank 8. After reading the file with the `read` command, the list of subgroups of a simple group `g` of rank not exceeding 8 is printed by the call `maxsub(g)`. The corresponding restriction matrix for branching is also available as `resmat(g,h,i)`, for branching to the `i`-th subgroup in the list that is of type `h`. When the integer `i` is omitted it is taken to be 1.

```
> read MAXSUB
> maxsub(E7)
    A2
    A1
    A1
    A1F4
```

```
     G2C3
     A1G2
     A1A1
> b=resmat(E7,A1G2)
> b
| 2 2 0 |
| 3 1 1 |
| 4 2 1 |
| 4 4 1 |
| 5 4 0 |
| 4 1 1 |
| 1 0 1 |

> branch([0,0,0,0,0,0,2],A1G2,b,E7)
| 4 1 1 1 |
| 6 2 0 1 |
| 2 0 2 1 |
| 4 2 0 1 |
| 2 1 1 1 |
| 0 3 0 1 |
| 4 0 1 1 |
| 2 2 0 2 |
| 4 1 0 1 |
| 6 0 0 1 |
| 2 1 0 1 |
| 0 0 1 1 |
| 0 1 0 1 |
| 2 0 0 1 |

>
```

Chapter 3. Algorithms

In this chapter some of the algorithms implemented in the LiE software package are treated in detail, in particular those that are needed for the calculation of tensor products and branch rules. We stress the underlying mathematical ideas, without reference to the actual C-code.

3.1. Weight algorithms

In this section the algorithms used in the calls `domweights` and `mul` are explained. All calculations take place in the dominant chamber of the weight system of a given representation.

3.1.1 The weight system. The algorithm used to calculate the dominant part of the weight system of a highest weight module is known as the Dynkin layer method (see Belinfante's article in [1]). It was published first in 1952 by E.B. Dynkin in the appendix of [7]. Roughly speaking, the procedure to determine all dominant weights is to start with the highest weight and then repeatedly subtract positive roots in order to obtain new dominant weights. A detailed description and proofs can be found in Moody & Patera [16].

3.1.2 Multiplicities. In order to compute the multiplicities of the dominant weights of a highest weight module, one can use Freudenthal's recursion formula, originally published in 1954 [8]. Many adaptations of this formula exist with a better performance on a computer; see Belinfante in [1] for an overview of some of the first implementations.

The implementation of Freudenthal's recursion formula in LiE is similar to that of Krusemeyer [12]. Using the fact that the multiplicities of weights in the same Weyl group orbit are the same, the recursion formula as in Humphreys [10] (see 1.2.5) can be written as follows: if the dominant representative of a weight w is denoted by $d(w)$, the multiplicity m_μ of the weight μ in the irreducible module of highest weight λ is given by

$$m_\mu = \frac{2}{c_\lambda - c_\mu} \sum_{\alpha \succ 0} \sum_{i \geq 0}^{\infty} (\mu + i\alpha, \alpha) m_{d(\mu + i\alpha)},$$

where $c_\mu = (\mu + \delta, \mu + \delta)$ with $\delta = \frac{1}{2} \sum_{\alpha \succ 0} \alpha$ (the summation over "$\alpha \succ 0$" being taken over all positive roots α). It follows that the calculation can be done keeping track of the multiplicities of dominant weights only. We know that m_μ is zero if $\mu \not\preceq \lambda$, and that $m_\lambda = 1$, so with this expression of m_μ for $\mu \prec \lambda$ in terms of the m_ν with ν dominant and $\mu \prec \nu \preceq \lambda$, all multiplicities can be calculated recursively. There is a simple (and fast) algorithm for the computation of $d(w)$ out of w, which is also used in the procedure `dominant` in LiE (again see Krusemeyer [12]).

More recently, Moody and Patera have published a version of Freudenthal's recursion formula that exploits the Weyl group symmetry even further using the following

observation: if two positive roots are conjugate under a Weyl group element leaving μ fixed, then their contributions to the right hand side of the above formula are equal. In [16] they show that for a suitable set $\{\zeta_1, \ldots, \zeta_t\}$ of positive representatives of the root system under the action of the group $\overline{\mathcal{W}}_\mu = \langle \mathcal{W}_\mu, -1 \rangle$, where $\mathcal{W}_\mu$ is the stabilizer of μ in the Weyl group, we have

$$m_\mu = \frac{1}{c_\lambda - c_\mu} \sum_{t=1}^{n} \#(\overline{\mathcal{W}}_\mu \zeta_t) \sum_{i \geq 0}^{\infty} (\mu + i\zeta_t, \zeta_t) m_{d(\mu + i\zeta_t)}.$$

This formula was used in the book [5] with multiplicity tables by Bremner, Moody and Patera, but in LiE this has not (yet?) been implemented, as the current algorithm is not the slowest link in the procedures for tensor product decompositions and branch rules.

3.2. Weyl group orbits

The version of Freudenthal's recursion formula used in LiE only involves dominant weights, but many calculations concerning representations use a summation over the whole Weyl group orbit of dominant weights. As the Weyl group can become very large, such a summation can be very time consuming, and it is often impossible to store an entire orbit. In many cases, like in the case of a summation over an orbit, it is not even necessary to store the entire orbit of a weight, and the program should just run quickly through the orbit and perform a particular task with each of the obtained weights.

As the orbit of a weight over a non-simple group can be obtained easily from the orbits of its components corresponding to the simple components of the group, we will assume that the group is simple.

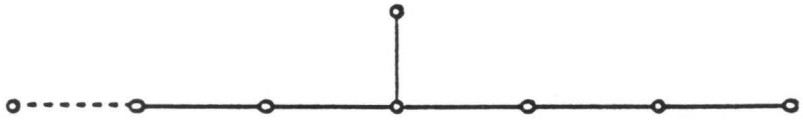
3.2.1 Classical groups. A particularly easy case is the case that the group is of type $\mathbf{A}_n$, because the Weyl group of $\mathbf{A}_n$ is isomorphic to the permutation group on $n+1$ letters. In fact, the root system can be embedded in the $n+1$-dimensional euclidean space $\mathbf{R}^{n+1}$ in such a way that the Weyl group acts by permutations of the coordinates. It follows that after a suitable linear transformation, the calculation of the Weyl group orbit can be done simply by generating all permutations of a given finite sequence of integers. Using lexicographical order, it is not difficult to run through all elements of the orbit using very little memory. The other classical groups ($\mathbf{B}_n$, $\mathbf{C}_n$ and $\mathbf{D}_n$) can be dealt with in a similar manner, where the Weyl group operates by permutations of coordinates and certain sign changes (see e.g. Humphreys [10], pages 64-65). Again we can use a lexicographical order to generate the orbit weight by weight.

3.2.2 Exceptional groups. There are only five finite groups to be dealt with, so in a theoretical sense the problem can be solved in constant time (if a basic arithmetical operation with integers takes constant time), but in practice these Weyl groups are too

large to deal with using a naive algorithm. For instance, the Weyl group of type $\mathbf{G}_2$ has 12 elements and for $\mathbf{F}_4$ the Weyl group has 1152 elements, which is not a problem for most computers, even with an inefficient algorithm, but the Weyl group of type $\mathbf{E}_8$ has 696729600 elements. Therefore our main concern is the Weyl group orbit calculation for type $\mathbf{E}_6$, $\mathbf{E}_7$ and $\mathbf{E}_8$, but we deal with $\mathbf{F}_4$ and $\mathbf{G}_2$ in the same manner.

The algorithm for the classical simple groups uses the explicit description of the Weyl group as the group of permutations of coordinates and suitable sign changes. In order to use use similar methods for the calculation of Weyl group orbits for the exceptional groups ($\mathbf{E}_6$, $\mathbf{E}_7$, $\mathbf{E}_8$, $\mathbf{F}_4$ and $\mathbf{G}_2$), detailed information on the structure of the Weyl groups is needed. The basic idea is to choose a large subgroup C of the Weyl group $\mathcal{W}$ that is isomorphic to a Weyl group of classical type and determine a complete set of left coset representatives $\sigma_1, \ldots, \sigma_n$ of C in $\mathcal{W}$. Suppose we want to generate the $\mathcal{W}$-orbit of a weight w. First we apply a suitable linear transformation in order to let the group C act by permutations (and possibly sign changes). Then we compute the set $\{w_1, \ldots, w_m\}$ of representatives of the C-orbits of $(w)\sigma_1, \ldots, (w)\sigma_n$ that are minimal with respect to lexicographical order (here we use the notation of the right action of $\mathcal{W}$ on the weight lattice, like in LiE). This means that $m \leq n = [\mathcal{W} : C]$ vectors need to be stored. Like in the classical case, the C-orbits of the weights $w_1, \ldots, w_m$ can now be generated consecutively, and these are guaranteed to be disjoint. As $\mathcal{W} = \cup_{i=1}^{n} \sigma_i C$, this procedure will encounter each weight in the $\mathcal{W}$-orbit of w exactly once. It remains to compute a large subgroup for each of the Weyl groups of type $\mathbf{E}$ (and $\mathbf{F}_4$ and $\mathbf{G}_2$), and a set of left coset representatives that enables us to compute the set $\{w_1, \ldots, w_t\}$ in a convenient way.

3.2.3 The Weyl group of $\mathbf{E}_7$. By way of example, we will treat the Weyl group $\mathcal{W}$ of $\mathbf{E}_7$ in more detail. The general procedure to find large classical subgroups is as follows: first write down the so-called extended Dynkin diagram. This is just the ordinary Dynkin diagram with one vertex added for minus the highest root and additional edges according to its inner product with the simple roots. For instance, the extended Dynkin diagram of $\mathbf{E}_7$ looks like this:



The dotted line indicates the additional edge. It follows that the highest root is the first fundamental weight, and that it spans a root system of type $\mathbf{A}_7$ with the simple roots $\alpha_1, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7$. The index of the Weyl subgroup is 72 (as can be inferred from the call `worder(E7)/worder(A7)` in LiE).

Next a set of 72 left coset representatives for the Weyl group of $\mathbf{A}_7$ in the Weyl group of $\mathbf{E}_7$ needs to be construct. As a first try, consider the Coxeter element $t = r_1 r_2 r_3 r_4 r_5 r_6 r_7$, where r_i is the reflection in the i-th simple root. The order of t is called the *Coxeter number* of $\mathbf{E}_7$, and it equals 18. It turns out that the 18 elements in the group $\langle t \rangle$ generated by t give rise to 18 distinct left cosets of the Weyl group of A_7. Using the Cayley computer program for group theoretical calculations A.M.

Cohen of the CWI in Amsterdam computed that for the element $c = t^{-1}r_5r_3r_4r_2$ the set $\langle t\rangle\{1, c, c^2, c^5\}$ consisting of the 72 products of the form t^jc^i with $i \in \{0, 1, 2, 5\}$ and $0 \leq j < 18$ constitutes a full set of representatives.

In 3.2.5 a Cayley program is included that was used to verify these results. The main step in the calculation involves the Todd Coxeter algorithm to construct the table of cosets, and to establish the action of the generators of the Weyl group of $\mathbf{E}_7$ as permutations of these cosets. The other exceptional groups can be dealt with in a similar way. The results are stated below.

3.2.4 The weyl groups of other exceptional types. The groups $\mathbf{G}_2$, $\mathbf{F}_4$, $\mathbf{E}_6$, and $\mathbf{E}_8$ have subgroups of type $\mathbf{A}_2$, $\mathbf{B}_4$, $\mathbf{D}_5$, and $\mathbf{D}_8$, as can be inferred from the extended Dynkin diagrams (see Bourbaki [3]). For $\mathbf{G}_2$ and $\mathbf{F}_4$ the Coxeter element furnishes sufficiently many coset representatives, but for $\mathbf{E}_6$, and $\mathbf{E}_8$ similar calculations as with $\mathbf{E}_7$ are needed. For $\mathbf{E}_6$ a maximal classical subgroup is obtained by leaving out the first vertex in the Dynkin diagram, so that the diagram of $\mathbf{D}_5$ remains. The index of the Weyl subgroup is 27, and the Coxeter number is 12. Denoting the Coxeter element by t, it turns out that the 36 elements $\langle t\rangle \cdot \{1, c, fc\}$ with $c = r_2r_4r_5r_6$ and $f = r_6r_4r_3$ will yield all 27 left cosets. In fact, the subset $\langle t\rangle \cdot \{fc\}$ may be replaced by $\langle t^4\rangle \cdot \{fc\}$ in order to obtain exactly 27 representatives.

Finally, it can be inferred from the extended Dynkin diagram of $\mathbf{E}_8$ that the Weyl group of $\mathbf{E}_8$ has a subgroup of type $\mathbf{D}_8$ and its index is 135. The Coxeter element t has order 30 and $t^{15} = -1$ lies in the Weyl group of $\mathbf{D}_8$, so we can find at most 15 cosets with $\langle t\rangle$. Again it can be shown with a Cayley script that for $S = \{1, c, c^2, c^3, c^4, c^5, d, d^3, d^4\}$ with $c = r_8r_6r_7r_5r_6r_3r_4r_5r_2r_4r_3r_1$ and $d = r_7r_4r_3r_2r_4r_5r_6r_1r_3r_4r_5r_2r_4r_3r_1$ the set $\langle t\rangle \cdot S$ provides the necessary $9 * 15 = 135$ left coset representatives.

3.2.5 Coset Computations in Cayley. For those who are familiar with the Cayley language the code that is necessary to verify the claims is included below for $\mathbf{E}_7$. The cases $\mathbf{E}_6$ and $\mathbf{E}_8$ were tested with Cayley in the same way.

In Cayley the Weyl group `WE7` of type $\mathbf{E}_7$ will be presented with generators and relations as in section 2.3, and the subgroup `A` is given by generators in terms of the generators of `WE6`. In LiE a simple `wword`-call gives a Weyl word for the reflection in the highest root: we first compute the matrix of the reflection with respect to the basis of simple roots, then we do a base change to the simple roots and call `wword`. The result can be verified with `weylmat`.

```
> r=reflection(highroot(E7),E7)
> rw=(icartan(E7)*r*cartan(E7))/detcartan(E7)
> s=wword(rw,E7)
> s
[ 1 3 4 5 6 2 4 5 3 4 1 3 2 4 5 6 7 6 5 4 3 2 4 5 6 1 3 4 5 2 4 3 1 ]
> if weylmat(s,E7)==r then print("OK") fi
OK
>
```

The next step in the Cayley program is the computation of a set of representatives of the right cosets of the subgroup by the Todd Coxeter algorithm, which also gives the index

of the subgroup (and we already know this should be 72). Next, a homomorphism is constructed (in the sense of the Cayley language), which gives the representation of the whole group as a permutation group on the right cosets. The coset defined be the identity element is denoted by 1. Finally, the proposed set of representatives in enumerated (note that they are the inverses of the elements given above, as we actually needed left cosets), and it is computed how many cosets they put together. If the number printed at the end is the index of the subgroup, then we have a full set of representatives. All these things are done in the following Cayley procedure:

```
     LIBRARY E7;
    we7: FREE(r1,r2,r3,r4,r5,r6,r7);
    we7.RELATIONS: r1^2=r2^2=r3^2=r4^2=r5^2=r6^2=r7^2=1,
  (r1*r2)^2=(r1*r4)^2=(r1*r5)^2=(r1*r6)^2=(r1*r7)^2=1,
  (r2*r3)^2=(r2*r5)^2=(r2*r6)^2=(r3*r6)^2=(r3*r5)^2=(r4*r6)^ 2=1,
  (r2*r7)^2=(r3*r7)^2=(r4*r7)^2=1,
  (r1*r3)^3=(r2*r4)^3=(r3*r4)^3=(r4*r5)^3=(r5*r6)^3=1,
  (r5*r7)^2=(r6*r7)^3=1;

s=r1*r3*r4*r5*r6*r2*r4*r5*r3*r4*r1*r3*r2*r4*r5*r6*r7*r6*r5*r4*r3*r2*r4*r5*
  r6*r1*r3*r4*r5*r2*r4*r3*r1;
a=<s,r1,r3,r4,r5,r6,r7>;
i,ct = TODD COXETER(we7,a);
PRINT 'Index of the subgroup:';
PRINT i;
phi,im,k = COSACT HOMOMORPHISM(WE7,A);
PRINT 'Image of the Weyl group of E7 in the permutation group on right cosets:';
PRINT im;

coxelt := r1*r2*r3*r4*r5*r6*r7;
coxperm=phi(coxelt);
coxeter=<coxperm>;
PRINT 'Subgroup C generated by the Coxeter element: ';
PRINT coxeter;
c=r2*r4*r3*r5*coxelt;
trans=[(IDENTITY OF we7),c,c^2,c^5];
ptrans=phi(trans);
PRINT 'Set S of four additional elements:';
PRINT ptrans;

orb=[1];
FOR EACH x IN ptrans DO
FOR EACH y IN coxeter DO
orb= orb JOIN [1^(x*y)];
END;
END;
PRINT 'Number of right cosets covered by SC:';
PRINT order(orb);


    FINISH;
```

The following Cayley session executes the above routine. Only the lines starting with ">" are typed in by the user. The calculation was done on a VAX-11/780 at CWI in Amsterdam.

```
VAX/UNIX CAYLEY     V3.5-1            29 Jun 89 19:39:32   STORAGE 1000000

 >set libfile='.';
 >library e7;


    Index of the subgroup:
    72
    Image of the Weyl group of E7 in the permutation group on right cosets:

GROUP IM
GENERATORS :
 (4,5)(6,14)(7,13)(8,21)(17,18)(19,22)(20,39)(23,27)(25,37)(26,38)(29,42)(30,43)
(41,44)(45,50)(46,51)(49,58)(53,59)(54,61)(62,65)(66,69)

 (1,2)(7,8)(9,12)(13,21)(19,23)(22,27)(24,28)(25,29)(26,30) (32,35)(33,36)(37,42)
(38,43)(41,46)(44,51)(48,52)(49,54)(55,57)(58,61)(71,72)

 (3,4)(7,12)(8,9)(14,15)(16,17)(19,24)(23,28)(25,35)(26,36)(29,32)(30,33)(39,40)
(41,48)(46,52)(49,57)(50,60)(54,55)(59,64)(65,68)(69,70)

 (2,3)(6,7)(9,10)(13,14)(17,22)(18,19)(20,26)(28,31)(29,45)(33,34)(35,47)(38,39)
(42,50)(46,53)(48,56)(51,59)(54,62)(57,63)(61,65)(70,71)

 (3,15)(4,14)(5,6)(10,11)(19,25)(22,37)(23,29)(24,35)(26,41)(27,42)(28,32)
(30,46)(33,52)(36,48)(38,44)(43,51)(62,66)(63,67)(65,69)(68,70051

 (6,18)(7,19)(8,23)(9,28)(10,31)(12,24)(13,22)(14,17)(15,16)(21,27)(41,49)
(44,58)(46,54)(48,57)(51,61)(52,55)(53,62)(56,63)(59,65)(64,68)

 (16,40)(17,39)(18,20)(19,26)(22,38)(23,30)(24,36)(25,41)(27, 43)(28,33)(29,46)
(31,34)(32,52)(35,48)(37,44)(42,51)(45,53)(47,56)(50,59)(60,64051

    Subgroup C generated by the Coxeter element:

GROUP COXETE IS A SUBGROUP OF IM
GENERATORS :
 (1,40,58,32,7,43,57,66,71,72,60,21,36,54,37,9,5,2)(3,39,49,69,45,64,27,48,62,
70,50,8,4,20,16,44,28,6)(10,33,18,15,38,31,52,19,51,63,35,53,68,42,56,24,46,22)
(11,34,55,25,59,23,14,26,61,67,47,12,30,17,41,65,29,13)

    Set S of four additional elements:
    [  IDENTITY,  (1,26,19,7,5,3)(2,40,49,71,60,8)(4,38,63,69,42,10)
(6,30,16,41,68,45)(9,43,58,50,48,17)(11,33,65,67,35,14)(12,34,22,56,61,32)
(13,36,55,47,51,31)(15,20,62,25,64,23)(18,53)(21,39,28,59,57,37)(24,52)(27,44)
(29,46,54,66,70,72),  (1,19,5)(2,49,60)(3,26,7)(4,63,42)(6,16,68)(8,40,71)
(9,58,48)(10,38,69)(11,65,35)(12,22,61)(13,55,51)(14,33,67)(15,62,64)(17,43,50)
```

[22] F. Warner, *Foundations of Differentiable Manifolds and Lie Groups*, Springer-Verlag GTM 94, New York, 1983. Originally published in 1971.

[23] W.C. Waterhouse, *Introduction to Affine Group Schemes*, Springer-Verlag GTM 66, New York, 1979.

[24] D.J. Winter, *Abstract Lie Algebras*, M.I.T. Press, Cambrigde, Massachusetts, 1972.

References

[1] R.E. Beck & B. Kolman (eds.), *Computers in Nonassociative Rings and Algebras*, Academic Press, New York, 1977.

[2] A. Borel & J. de Siebenthal, *Les sous-groupes fermés de rang maximum des groupes de Lie clos*, Commentarii Math. Helv. **23** (1949), pp. 200-221.

[3] N. Bourbaki, *Groupes et algèbres de Lie, Chap 4, 5, et 6*, Hermann, Paris, 1968.

[4] N. Bourbaki, *Groupes et algèbres de Lie, Chap 7 et 8*, Hermann, Paris, 1975.

[5] M.R. Bremner, R.V. Moody, J. Patera, *Tables of dominant weight multiplicities for representations of simple Lie algebras*, Dekker, Monographs and Textbooks in Pure and Appl. Math. 90, New York, 1985.

[6] T. Bröcker & T. tom Dieck, *Representations of Compact Lie Groups*, Springer-Verlag GTM 98, New York, 1985.

[7] E.B. Dynkin, *The structure of semisimple algebras*, Uspehi Nat. Nauk (N.S.), **2** (1947), pp. 59-127.

[8] H. Freudenthal, *On the calculation of the characters ois semisimple Lie groups, II*, Indag. Math, **16** (1954), pp. 487-491.

[9] G. Hochschild, *The Structure of Lie Groups*, Holden-Day, San Francisco, 1965.

[10] J. Humphreys, *Introduction to Lie Algebras and Representation Theory*, Springer-Verlag GTM 9, New York, 1972.

[11] J. Humphreys, *Linear Algebraic Groups*, Springer-Verlag GTM 21, New York, 1975.

[12] M.I. Krusemeyer, *Determining Multiplicities of Dominant Weights in Irreducible Lie Algebra Representations Using a Computer*, BIT **11** (1971), pp.310-316.

[13] **LiE** *manual*, Computer Algebra Group of the Centre for Mathematics and Computer Science, Amsterdam, 1989.

[14] W.G. McKay & J. Patera, *Tables of dimensions, indices and branching rules for representations of simple Lie algebras*, Dekker, Lecture Notes in Pure and Appl. Math. 69, New York, 1981.

[15] D. Montgomery & L. Zippin, *Topological transformation groups*, Interscience Publishers, New York, 1955.

[16] R.V. Moody & J. Patera, *Fast Recursion Formula for weight Multiplicities*, Bull. Am. Math. Soc. **7** (1982), pp. 237-242.

[17] L.S. Pontrjagin, *Topological Groups*, Princeton University Press, Princeton, 1939.

[18] J.-P. Serre, *Algèbres de Lie semi-simples complexes*, Benjamin, New York, 1966.

[19] T.A. Springer, *Linear Algebraic Groups*, Birkhäuser, Boston, 1981.

[20] J. Tits, *Tabellen zu den einfachen Lie Gruppen und ihren Darstellungen*, Springer, Lecture Notes in Math. 40, Berlin, 1967.

[21] V.S. Varadarajan, *Lie Groups, Lie Algebras, and Their Representations*, Springer-Verlag GTM 102, New York, 1984. Originally published in 1974.

each dominant weight in the weight system, and for every encountered weight λ' its contribution is added to the decomposition matrix under construction. It remains to define the contribution of a weight λ' of V. Let $m_V(\lambda')$ be the multiplicity of λ in V, and let $\sigma \in \mathcal{W}$ such that $\sigma(\lambda'+\mu+\delta)$ is the dominant representative of the Weyl group orbit of $\lambda'+\mu+\delta$. There wil be no contribution of λ' if the weight $w = \sigma(\lambda'+\mu+\delta)-\delta$ is not dominant, but if it is, the weight w will be added with multiplicity $\epsilon(\sigma)m_V(\lambda')$, where $\epsilon(\sigma)$ is the sign (i.e. the determinant) of σ (neither w nor $\epsilon(\sigma)$ depend upon the choice of the element $\sigma \in \mathcal{W}$).

Considerable effort was made to make the implementation of this algorithm in LiE efficient. For instance, a suitable data structure was developed in order to insert the contributions quickly (instead of collecting them in one big matrix and applying `redmulmat`). Of course the summation is done over the smallest of the two weight systems.

3.3.2 Branching. As in 2.5.3 let G_1 be a subgroup of a group G_2 with maximal tori $T_1 \subset T_2$, and suppose that the restriction map $\Lambda_2 \to \Lambda_1$ of the weight lattice Λ_2 of G_2 to the weight lattice Λ_1 of G_1 is known. Furthermore let V be a module over G_2 whose restriction to G_1 we want to decompose into irreducible G_1-modules. For $x \in T_2$, $\lambda \in \Lambda_2$ and $v \in V_\lambda$, we have $xv = \lambda(x) \cdot v$, whence $xv = (\lambda|_{T_1})(x) \cdot v$ for $x \in T_1 \subset T_2$. It follows that the weights of the G_1-module V are just the restrictions of the weights of V as a G_2-module and that the multiplicity matrix of V over G_1 can be obtained from the multiplicity matrix of V as a G_2-module as follows: generate all weights of V over G_2 (not just the dominant ones) with their multiplicities, and restrict them to T_1 (leaving multiplicities unchanged); then select the weights that are dominant for G_1.

This is exactly what is done in LiE : `mul` is called, and the resulting multiplicity matrix is stored, then for each dominant weight the procedure to run through the orbit of that weight is called and for every weight obtained the restriction to T_1 is calculated. If this restricted weight is dominant then it is inserted in the multiplicity matrix of the restricted module that is being constructed, and otherwise we go to the next weight in the orbit. Finally, the function `decomp` is called in order to get the decomposition matrix of the restricted module.

Of course this algorithm can be improved considerably if there is a way to predict whether a weight of G_2 will become dominant over G_1. For instance, for the branching from $\mathbf{F}_4$ to $\mathbf{B}_4$ we only need to generate 3 chambers in the weight system for $\mathbf{F}_4$ in order to find all dominant induced weights of $\mathbf{B}_4$, and not 1152. So if a user needs to do branching in a specific situation where more is known about the relative position of the two Weyl groups, it is probably wise to write a special program for this purpose. The language of LiE (see the manual [13]) provides a useful environment for the development of such programs.

```
(20,25,23)(21,28,57)(29,54,70)(30,41,45)(31,36,47)(32,34,56)(37,39,59)(46, 66,72)
,  (1,3,5,7,19,26)(2,8,60,71,49,40)(4,10,42,69,63,38)(6,45,68,41,16,30)
(9,17,48,50,58,43)(11,14,35,67,65,33)(12,32,61,56,22,34)(13,31,51,47,55,36)
(15,23,64,25,62,20)(18,53)(21,37,57,59,28,39)(24,52)(27,44)(29,72,70,66,54,46) ]
    Number of right cosets covered by SC:
    72
 >quit;

     END OF RUN.
  24.850 SECONDS,  999986 WORDS USED.
```

3.3. Tensor products and branch rules

In this section, the algorithms used in the functions `tensor` and `branch` of LiE are explained. The algorithms used work for all (simple) Lie groups. The main ingredient in both is the procedure to run through the Weyl group orbit of a weight. There are more subtle algorithms with a better performance that use Young tableaux, but as yet they have not been made to comprise all simple Lie groups. There are plans to develop and implement new algorithms in LiE in the next few years.

3.3.1 Klimyk's formula. Let V and W be two modules over a group G. The tensor product $V \otimes W$ of V and W is defined to be the tensor product of V and W as vector spaces, with the action $g(x \otimes y) = g(x) \otimes g(y)$ for $g \in G$, $x \in V$ and $y \in W$. Having fixed a maximal torus T of G, it follows that for weights $\lambda, \mu \in \Lambda$ of G, the action of T on the tensor product $V_\lambda \otimes W_\mu$ of weight spaces is given by the weight $\lambda + \mu$. Therefore, the multiplicity matrix of $V \otimes W$ can be computed as follows: for each weight λ of V and μ of W with multiplicities $m_V(\lambda)$ and $m_W(\mu)$ add the weight $\lambda + \mu$ to the multiplicity matrix under construction with multiplicity $m_V(\lambda) * m_W(\mu)$ if $\lambda + \mu$ is dominant. The decomposition matrix can then be found by calling the procedure `decomp`. Obviously, this algorithm can be improved by letting λ run over the dominant weights of V only, and replacing $\lambda + \mu$ by the dominant representative $d(\lambda + \mu)$ of its Weyl group orbit. But we still need a summation over the Weyl group and a `decomp` call this way.

In order to apply a more efficient algorithm, first note that it suffices to find a procedure that computes the tensor product of two highest weight modules V and W, say given by weights λ and μ. The algorithm above involves two nested summations: over the dominant weights of V and over all weights of W. We now give an algorithm of A.U. Klimyk that produces a decomposition matrix directly (see Humphreys [10] , p.142, exercise 9). The algorithm only uses one summation over all weights of one module.

For each weight λ' in the weight system of V, we will define its contribution to the decomposition matrix of the tensor product. The first step of the algorithm is to compute the dominant part of the weight system of V with multiplicities using the function `mul`. Then the procedure to run through the Weyl group orbit is called for