Computing in reductive groups

Bill Casselman
University of British Columbia
cass@math.ubc.ca

Suppose G to be a split group defined over a field F of characteristic 0. An element of G possesses a canonical form incorporating the Bruhat decomposition. In these terms, how can one do multiplication in the group? I. e. given the Bruhat normal forms of g and h, how can one find that of gh? Given the normal form of g, what is that of g^{-1}?

These questions are answered clearly in [Cohen-Murray-Taylor:2005]. This has been incorporated in widely used computer programs. One starts with a good basis of the Lie algebra, and assumes that one knows how to compute Lie brackets. One also assumes that one knows how to apply elements of the normalizer of a maximal torus to elements of the Lie algebra. These are explained in [Casselman:2018], among other places. The original problem then reduces to finding formulas for the commutator of two unipotent elements. This was done a long time ago by Chevalley, and explained reasonably well in [Carter:1972]. But the arguments depend on special tricks. There is no systematic technique. The situation is not completely satisfying.

The book [Steinberg:1967] suggests a systematic approach, and in this note I'll show how it leads to a practical algorithm. One might hope that this would lead also to something new, say for Kac-Moody groups, but as [Morita:1987] points out there is nothing new to appear for these. The only value of this note therefore appears to be aesthetic.

Contents

1. Normal form

Suppose N to be an arbitrary unipotent algebraic group, possessing a filtration

$$N = N_0 \supset N_1 \supset \ldots \supset N_{n-1} \supset N_n = \{1\}\,,$$

in which each N_i/N_{i+1} is one dimensional, and in the centre of N/N_{i+1}. I assume also that for each i we are given an embedding $\mathfrak{e}_i$ of F into N_i, inducing an isomorphism with N_i/N_{i+1}. Let $N_{[i]}$ be the image. As an immediate consequence of assumptions:

1.1. Lemma. *For every i, j the commutator $(\mathfrak{e}_i(x), \mathfrak{e}_j(y))$ lies in $N_{1+\max i,j}$.*

With consequence:

1.2. Proposition. *Every element in N_k has a* **normal form**

$$\nu = \prod_0^{n-1} \mathfrak{e}_i(x_i) \quad (\textit{increasing indices left to right}) \tag{1.3}$$

inducing an isomorphism of the algebraic variety N with F^n.

Proof. An easy induction.

The basic data defining the group structure are commutation formulas

$$\mathfrak{e}_j(y)\mathfrak{e}_i(x) = \mathfrak{e}_i(x)\mathfrak{e}_j(y) \prod_{k>\max(i,j)} \mathfrak{e}_k(C^k_{i,j}(x,y)) \quad (i<j)\,. \tag{1.4}$$

(again in increasing order) with $C_{i,j}^k$ a polynomial in x, y and all $k > \max(i,j)$. If N is the unipotent radical of a minimal parabolic subgroup in a semi-simple group, one can find a filtration of it compatible with the heights of roots, and each $\mathfrak{e}_i$ the exponential of a root space in the Lie algebra.

Such expressions can be used in an algorithm to find the normal form of any element of N. The input will be a product of elements $\mathfrak{e}_i(x_i)$. The basic idea is suggested by the inductive proof above. (1) First locate the last occurrence of $\mathfrak{e}_0$ in the product. Then keep moving it to the left, using commutations. At any moment, we shall be in one of three situations: (1) The term $\mathfrak{e}_0$ will be at the far left. We stop and go on to deal with $\mathfrak{e}_1$, as I'll discuss in a moment. (2) Or there is an expression $\mathfrak{e}_i$ just to the left of $\mathfrak{e}_0$. We are looking at $\mathfrak{e}_i(x)\mathfrak{e}_0(y)$, with one of two possibilities: $i = 0$ or $i > 0$. In the first case replace the two-term product by $\mathfrak{e}_0(x+y)$. In the second we apply a commutation relation, and replace the two-term product by an expression $\mathfrak{e}_0(y)\mathfrak{e}_i(x)n$, with n in N_1. We continue moving $\mathfrak{e}_0$ to the left in this way. After we have moved $\mathfrak{e}_0$ all way to the left, we are looking at some $\mathfrak{e}_0(x)n$ with n in N_1. We know apply the induction hypothesis, and carry on with n in the same way, replacing $\mathfrak{e}_0$ with $\mathfrak{e}_1$. Etc.

I'd like to point out that this computation requires only knowing the commutator of two elementary unipotent matrices.

Example. Suppose $\mathfrak{n}$ to be the vector space of upper triangular nilpotent matrices in M_n. For $1 \leq i < j \leq n$, let $e_{i,j}$ be the matrix in $\mathfrak{n}$ with all entries equal to 0 except that at (i,j), which is 1. These form a linear basis of $\mathfrak{n}$.

Let $\mathfrak{a}$ be the vector space of diagonal matrices (a_i) in M_n, ε_i the linear function on $\mathfrak{a}$ taking a to a_i. It is a Lie algebra. Every a in $\mathfrak{a}$ corresponds to the adjoint operator

$$\mathrm{ad}_a\colon\ e \longmapsto [a,e] = ae - ea\,.$$

The space $\mathfrak{n}$ is stable under ad_a, and each $e_{i,j}$ is an eigenvector for all of $\mathfrak{a}$:

$$\mathrm{ad}_a(e_{i,j}) = (a_i - a_j)e_{i,j} = \langle \varepsilon_i - \varepsilon_j, a\rangle e_{i.j}\,.$$

The functions

$$\alpha_i = \varepsilon_i - \varepsilon_{i+1}$$

for $1 \leq i < n$ are a basis of the space spanned by these eigencharacters. In particular, if $i < j$ then

$$\varepsilon_i - \varepsilon_j = \alpha_i + \cdots + \alpha_{j-1}\,.$$

The basic formula here is the matrix multiplication

$$e_{i,j}\cdot e_{k,\ell} = \begin{cases} e_{i,\ell} & \text{if } j = k \\ e_{k,j} & \text{if } i = \ell \\ 0 & \text{otherwise} \end{cases}\quad.$$

This leads to the Lie bracket commutation relation

$$[e_{i,j}, e_{k,\ell}] = \begin{cases} e_{i,\ell} & \text{if } j = k \text{ (and } i < k) \\ -e_{k,j} & \text{if } i = \ell \text{ (and } k < i) \\ 0 & \text{otherwise.} \end{cases}$$

The two cases can be phrased in terms of roots. In the first, for example, with $i < k$, then

$$\varepsilon_i - \varepsilon_j = \alpha_i + \cdots + \alpha_{j-1}, \varepsilon_k - \varepsilon_\ell = \alpha_i + \cdots + \alpha_{\ell-1}\,.$$

So we can formulate these equations as

$$[e_\lambda, e_\mu] == \pm e_{\lambda+\mu}$$

where the sign is determined by whether $\lambda < \mu$ $(+)$ or $\lambda > \mu$ $(-)$, in some obvious sense.

The matrix product of two matrices in $\mathfrak{n}$ is again in $\mathfrak{n}$ (but might vanish). If e and f are eigenvectors for the adjoint action, say with eigencharacters λ, μ, so is $e \cdot f$, with eigencharacter $\lambda + \mu$.

1.5. Lemma. *Suppose each n_i to be in $\mathfrak{n}$. The matrix product $\prod n_i$ vanishes if two or more of the n_i coincide.*

Proof. We may as well assume each n_i to be an eigenvector for the adjoint action. For $i < j$

$$\varepsilon_i - \varepsilon_j = \alpha_i + \cdots + \alpha_{j-1}\,.$$

There are no scalars other then 0 or 1.

Every nilpotent matrix x in corresponds to a unipotent matrix $I + x$.

1.6. Proposition. *If x, y are matrices in $\mathfrak{n}$, then we have the commutator formula*

$$(I+x)^{-1}(I+y)^{-1}(I+x)(I+y) = I + [x,y]\,.$$

The left-hand side is

$$\begin{aligned}(I-x-y+xy)(I+x+y+xy) = I &- x - y + xy\\ &x - x^2 - yx + xyx\\ &y - xy - y^2 + xy^2\\ &xy - xyx - yxy + xyxy\,.\end{aligned}$$

According to the Lemma, many terms vanish, and there are also some cancellations.

Example. Suppose $G = \mathrm{GL}_3$ with positive roots $\alpha < \beta < \alpha + \beta$. If

$$\mathfrak{e}_\alpha(x) = \begin{bmatrix} 1 & x & \circ \\ \circ & 1 & \circ \\ \circ & \circ & 1 \end{bmatrix}, \quad \mathfrak{e}_\beta(x) = \begin{bmatrix} 1 & \circ & \circ \\ \circ & 1 & x \\ \circ & \circ & 1 \end{bmatrix}, \quad \mathfrak{e}_{\alpha+\beta}(x) = \begin{bmatrix} 1 & \circ & x \\ \circ & 1 & \circ \\ \circ & \circ & 1 \end{bmatrix}$$

then

$$\mathfrak{e}_\beta^{-1}(y)\mathfrak{e}_\alpha^{-1}(x)\mathfrak{e}_\beta(y)\mathfrak{e}_\alpha(x) = \begin{bmatrix} 1 & \circ & -xy \\ \circ & 1 & \circ \\ \circ & \circ & 1 \end{bmatrix}$$

or

$$\mathfrak{e}_\beta(y)\mathfrak{e}_\alpha(x) = \mathfrak{e}_\alpha(x)\mathfrak{e}_\beta(y)\mathfrak{e}_{\alpha+\beta}(-xy)\,. \tag{1.7}$$

Example. Suppose $G = \mathrm{Sp}(4)$ with positive roots

$$\alpha < \beta < \alpha + \beta < 2\alpha + \beta\,.$$

If

$$\mathfrak{e}_\alpha(x) = \begin{bmatrix} 1 & x & \circ & \circ \\ \circ & 1 & \circ & \circ \\ \circ & \circ & 1 & -x \\ \circ & \circ & \circ & 1 \end{bmatrix}, \quad \mathfrak{e}_\beta(x) = \begin{bmatrix} 1 & \circ & \circ & \circ \\ \circ & 1 & x & \circ \\ \circ & \circ & 1 & \circ \\ \circ & \circ & \circ & 1 \end{bmatrix}$$

$$\mathfrak{e}_{\alpha+\beta}(x) = \begin{bmatrix} 1 & \circ & x & \circ \\ \circ & 1 & \circ & x \\ \circ & \circ & 1 & \circ \\ \circ & \circ & \circ & 1 \end{bmatrix}, \quad \mathfrak{e}_{2\alpha+\beta}(x) = \begin{bmatrix} 1 & \circ & \circ & x \\ \circ & 1 & \circ & \circ \\ \circ & \circ & 1 & \circ \\ \circ & \circ & \circ & 1 \end{bmatrix}$$

then

$$\mathfrak{e}_\beta^{-1}(y)\mathfrak{e}_\alpha^{-1}(x)\mathfrak{e}_\beta(y)\mathfrak{e}_\alpha(x) = \begin{bmatrix} 1 & \circ & -xy & x^2y \\ \circ & 1 & \circ & -xy \\ \circ & \circ & 1 & \circ \\ \circ & \circ & \circ & 1 \end{bmatrix}$$

or

(1.8) $$\mathfrak{e}_\beta(y)\mathfrak{e}_\alpha(x) = \mathfrak{e}_\alpha(x)\mathfrak{e}_\beta(y)\mathfrak{e}_{\alpha+\beta}(-xy)\mathfrak{e}_{2\alpha+\beta}(x^2y)\,.$$

$\circ$ ———— $\circ$

Example. I'll show how the normal form algorithm goes in a simple example. Suppose $n = 4$. The ordered list of root indices could be

$$(0,1),\ (1,2),\ (2,3),\ (0,2),\ (1,3),\ (0,3)\,.$$

This is how things go in finding the normal form of $\mathfrak{e}_{1,3}(1)\cdot\mathfrak{e}_{1,2}(1)\cdot\mathfrak{e}_{2,3}(1)\cdot\mathfrak{e}_{1,2}(1)$

shifting $(0,1)$:
shifting $(1,2)$:
$\mathfrak{e}_{1,3}(1)\cdot\mathfrak{e}_{1,2}(1)\cdot\mathfrak{e}_{2,3}(1)\cdot\mathfrak{e}_{1,2}(1)$
$\mathfrak{e}_{1,3}(1)\cdot\mathfrak{e}_{1,2}(1)\cdot\mathfrak{e}_{2,3}(1)\cdot\mathfrak{e}_{1,3}(-1)$
$\mathfrak{e}_{1,3}(1)\cdot\mathfrak{e}_{1,2}(2)\cdot\mathfrak{e}_{2,3}(1)\cdot\mathfrak{e}_{1,3}(-1)$
$\mathfrak{e}_{1,2}(2)\cdot\mathfrak{e}_{1,3}(1)\cdot\mathfrak{e}_{2,3}(1)\cdot\mathfrak{e}_{1,3}(-1)$
shifting $(2,3)$:
$\mathfrak{e}_{1,2}(2)\cdot\mathfrak{e}_{1,3}(1)\cdot\mathfrak{e}_{2,3}(1)\cdot\mathfrak{e}_{1,3}(-1)$
$\mathfrak{e}_{1,2}(2)\cdot\mathfrak{e}_{2,3}(1)\cdot\mathfrak{e}_{1,3}(1)\cdot\mathfrak{e}_{1,3}(-1)$
shifting $(0,2)$:
shifting $(1,3)$:
$\mathfrak{e}_{1,2}(2)\cdot\mathfrak{e}_{2,3}(1)\cdot\mathfrak{e}_{1,3}(1)\cdot\mathfrak{e}_{1,3}(-1)$
$\mathfrak{e}_{1,2}(2)\cdot\mathfrak{e}_{2,3}(1)\cdot$

Of course the last item doesn't have to be shifted.

2. Factoring N

We now know how to find the normal form of any element in N. This might, however, leave you slightly uneasy, since there is some arbitrariness involved. All depends on a choice of order on the roots. In fact, *we can write elements of N in any prescribed order.* In the literature, the following is usually left as an exercise, perhaps because the proof, although basically straightforward, requires some comment.

2.1. Proposition. *Any element of N can be written as a unique product of elements in the $N_{[i]}$ in any order.*

Proof. I'll explain an algorithm that proves this constructively. We start with an element of N expressed in normal form:

$$\nu = x_0 \dots x_{n-1}$$

with each x_i in $N_{[i]}$. We are also given a permutation σ of $[0,n)$, and at the end we want to express ν as a product

$$\nu = \prod y_{\sigma(i)} \quad \text{(ascending order of } i)\,.$$

in which, as before, y_i is in $N_{[i]}$.

The algorithm proceeds in n steps. At the m-th step the original element is expressed as a product

$$\nu = \nu_{<m} \cdot \nu_{\geq m}$$

in which $\nu_{<m}$ is a product of elements in the $N_{[i]}$ for $i < m$, in an order I'll specify in a moment, and $\nu_{\geq m}$ is an element of N_m expressed in normal form.

What order is $\nu_{<m}$ to be expressed in? Well, σ determines by restriction an order of the interval $[0, m-1]$. We get this by writing out the array of all $\sigma(i)$ and then deleting those not in $[0, m-1]$. For example, if $N = 5$ and the final order is 31042, the sequence of restricted orders is

$$\begin{gathered} ..0.. \\ .10.. \\ .10.2 \\ 310.2 \\ 31042\,. \end{gathered}$$

• To start, we just remove x_0 from $\nu_{\geq 0}$, making $\nu_{\geq 1} = x_1 \dots x_{n-1}$, and set $\nu_{<1} = x_0$. Of course we still have

$$\nu = x_0 \cdot x_1 \dots x_{n-1} = \nu_{<1} \cdot \nu_{\geq 1}\,.$$

• Next, we remove x_1 from $\nu_{\geq 1}$ and add it to $\nu_{<1}$. How we do this depends on σ. If 1 comes after 0 in the given order, we just tack x_1 onto the end of $\nu_{<1}$, making

$$\nu_{<2} = x_0 x_1\,,$$

and set $\nu_{\geq 2} = x_2 \dots x_{n-1}$. Otherwise, we write

$$x_0 x_1 = x_1 x_0 \cdot \gamma(x_0, x_1)\,,$$

in which $\gamma(a, b)$ is the commutator $a^{-1}b^{-1}ab$. The commutator in this expression is in N_2, so now we set

$$\nu_{\geq 2} = \gamma(x_0, x_1) x_2 \dots x_{n-1}\,.$$

We express this in normal order, which I'll write again as a product of x_i, and move onto step 3.

• In step m, we remove x_m from $\nu_{\geq m}$ and place it temporarily at the end of $\nu_{<m}$, getting

$$\nu_{<m} x_m\,.$$

We then move x_m into a place determined by the restricted order of degree m, applying a commutator relation. Thus

$$\nu_{<m} = y_0 \dots y_k \cdot y_{k+1} \dots y_{m-1}$$

becomes

$$\nu_{<m+1} = y_0 \dots y_k \cdot x_m \cdot y_{k+1} \dots y_{m-1}$$

We then tack the commutator $\gamma(y_{k+1} \dots y_{n-1}, x_m)$ onto the front of the temporary $\nu_{\geq m+1}$, getting the new $\nu_{\geq m+1}$.

The application of the commutator relation here is not as simple as it was in the normal form algorithm. In this process, the following lemma will be useful in induction.

2.2. Lemma. *Suppose a, b in N_i, N_j, x in N_k with $i, j < k$. Then*

$$\gamma(ab, x) = \gamma(a, x) \cdot \gamma(\gamma(a, x), b) \cdot \gamma(b, x)\,.$$

lies in N_{k+1}.

Proof. Because

$$\begin{aligned} ab \cdot x &= axb\gamma(b, x) \\ &= xa\gamma(a, x)b\gamma(b, x) \\ &= x \cdot ab \cdot \gamma(a, x)\gamma(\gamma(a, x), b)\gamma(b, x)\,. \end{aligned}$$

In summary, the construction goes by induction on the length of N, but it's not simple to describe clearly.

3. Chevalley's commutation formula

Again suppose G to be a semi-simple split group defined over the field F. Fix $(B, T, \{\mathfrak{x}_\alpha\})$, a frame of G with $B = TU$. Let Σ be the associated root system, Δ the simple roots. To the frame is associated an 'opposition involution' θ of $\mathfrak{g}$ that takes T to itself and $\mathfrak{g}_\lambda$ to $\mathfrak{g}_{-\lambda}$. Choose elements $\mathfrak{x}_\lambda$ of $\mathfrak{g}_\lambda$ such that $\mathfrak{x}_\lambda^\theta = \pm\mathfrak{x}_{-\lambda}$. These are determined uniquely by the frame up to sign. Together with the basis $([\mathfrak{x}_{-\alpha}, \mathfrak{x}_\alpha])_{\alpha\in\Delta}$ of $\mathfrak{t}$ they form a **Chevalley basis** of $\mathfrak{g}$. The map $t \mapsto t\mathfrak{x}_\lambda$ is an isomorphism of F with $\mathfrak{g}_\lambda$.

Recall that if λ and μ are roots then the λ-string through μ is the collection of roots of the form $\mu + m\lambda$:

$$\mu - p_{\lambda,\mu}\lambda, \cdots, \mu + q_{\lambda,\mu}\lambda\,.$$

If λ, μ, and $\lambda + \mu$ are all roots, then according to a formula of Chevalley

$$[\mathfrak{x}_\lambda, \mathfrak{x}_\mu] = N_{\lambda,\mu}\mathfrak{x}_{\lambda+\mu}\,.$$

with $|N_{\lambda,\mu}| = p_{\lambda,\mu} + 1$. Determining the sign of this structure constant is not at all trivial, but algorithms are known. (See [Cohen-Murray-Taylor:2005] and [Casselman:2018].)

For every unipotent group U there exists an algebraic exponential map $\exp$ from its Lie algebra to the group. It is functorial, in the sense that if ρ is a homomorphism of U (whose image is necessarily also unipotent) then

$$\exp(d\rho(x)) = \rho(\exp(x))\,.$$

In particular, $\mathrm{Ad}_{\exp(x)} = \exp(\mathrm{ad}_x)$. Now

$$\begin{aligned}[\mathfrak{x}_\lambda, \mathfrak{x}_\mu] &= N_{\lambda,\mu}\mathfrak{x}_{\lambda+\mu}\\ [\mathfrak{x}_\lambda, [\mathfrak{x}_\lambda, \mathfrak{x}_\mu]] &= N_{\lambda,\mu}N_{\lambda,\lambda+\mu}\mathfrak{x}_{2\lambda+\mu}\\ &\cdots\end{aligned}$$

so that if

$$M_{\lambda,\mu,m} = \frac{N_{\lambda,\mu}N_{\lambda,\lambda+\mu}\cdots N_{\lambda,(m-1)\lambda+\mu}}{m!}$$

then

$$\frac{\mathrm{ad}^m_{\mathfrak{x}_\lambda}(\mathfrak{x}_\mu)}{m!} = M_{\lambda,\mu,m}\mathfrak{x}_{m\lambda+\mu}\,.$$

if $m\lambda + \mu$ is a root. Note that $M_{\lambda,\mu,m}$ is always an integer, because of Chevalley's theorem about $|N_{\lambda,\mu}|$.

These are most efficiently computed by induction on m, so it is best to compute the whole sequence of $M_{\lambda,\mu,m}$ such that $m\lambda + \mu$ is a root. (They are all in the λ-string through μ.)

As a consequence:

$$\exp(u\mathfrak{x}_\lambda)\cdot v\mathfrak{x}_\mu\cdot\exp(-u\mathfrak{x}_\lambda) = \sum\nolimits_{m\ge0} M_{\lambda,\mu,m}\cdot u^m v\mathfrak{x}_{m\lambda+\mu}\,,$$

which we can rewrite as a commutation formula

$$\begin{aligned}\exp(u\mathfrak{x}_\lambda)\cdot v\mathfrak{x}_\mu &= \Big(\sum\nolimits_{m\ge0} M_{\lambda,\mu,m}\cdot u^m v\mathfrak{x}_{m\lambda+\mu}\Big)\exp(u\mathfrak{x}_\lambda)\\ &= \sum\nolimits_{m\ge0} M_{\lambda,\mu,m}\cdot u^m v\mathfrak{x}_{m\lambda+\mu}\exp(u\mathfrak{x}_\lambda)\,.\end{aligned} \tag{3.1}$$

The exponential map is canonical in a technical sense. If θ is an automorphism of U, it induces an automorphism of $\mathfrak{u}$, which I'll also call θ. Then

$$\theta(\exp(x)) = \exp(\theta(x))\,.$$

If θ is conjugation by $\exp(t\mathfrak{x}_\lambda)$ then by (3.1) this gives us

$$\begin{aligned}\exp(u\mathfrak{x}_\lambda)\exp(v\mathfrak{x}_\mu)\exp(-u\mathfrak{x}_\lambda) &= \exp\big(\exp(u\mathfrak{x}_\lambda)v\mathfrak{x}_\mu\exp(-u\mathfrak{x}_\lambda)\big)\\ &= \exp\Big(\sum_{m\geq 0} M_{\lambda,\mu,m}\cdot u^m v\mathfrak{x}_{m\lambda+\mu}\Big)\,.\end{aligned}\tag{3.2}$$

From now on, I'll write $\mathfrak{e}$ for the exponential map, and

$$\mathfrak{e}_\lambda\colon\ u\longmapsto \mathfrak{e}(u\mathfrak{x}_\lambda)=\exp(u\mathfrak{x}_\lambda)$$

for the embedding of F into G associated to the root λ.

Define the **closure** of the pair $\{\lambda,\mu\}$ to be the set of all roots of the form $k\lambda+\ell\mu$ with $k\geq 0$, $\ell\geq 0$ and $k+\ell>0$, and let its **strict closure** be the complement in this of $\{\lambda,\mu\}$. The discussion in §1 has as consequence a commutation formula

$$\mathfrak{e}_\mu(v)\mathfrak{e}_\lambda(u)=\mathfrak{e}_\lambda(u)\mathfrak{e}_\mu(v)\prod_{k,\ell>0}\mathfrak{e}_{k\lambda+\ell\mu}(c_{k,\ell}^{\lambda,\mu}\,u^k v^\ell)$$

for some constants $c_{k,\ell}^{\lambda,\mu}$, in which the product is over the strict closure of $\{\lambda,\mu\}$. One may assume the product to be ordered compatibly with the sums $k+\ell$. (We have already seen an example in (1.8) .) There is *a priori* some possible ambiguity in the order, but it can be empirically observed, in finite-dimensional Lie algebras, that this does not matter. Any two roots can be transformed to a pair embedded in a system of rank two associated to some couple of elements in Δ. It therefore suffices to consider systems of rank two.

We now ask:

What are the coefficients $c_{k,\ell}^{\lambda,\mu}$?

4. Steinberg's algorithm

A number of *ad hoc* methods allow you to go through all the finite-dimensional root systems of rank two case by case in order to compute the constants in Chevalley's formula. This is what one finds in the literature (for example, Chapter 5 of [Carter:1972]) , and I'll give some simple examples later. But for G_2 this is a painful business, and for that reason it is more interesting to apply an algorithm suggested by an argument presented in §3 of [Steinberg:1967]. With this, the $c_{k,\ell}=c_{k,\ell}^{\lambda,\mu}$ are found inductively. Steinberg's technique has the advantage that it applies uniformly in all cases, even Kac-Moody groups, for which Chevalley's formula regarding $|N_{\lambda,\mu}|$ remains valid. It is a bit too complicated to carry out by hand, but well suited to machine computation. One other drawback is that the formulas it produces are longer than those found in the literature, but that is probably easy to correct by examining them. The shortest formulas for G_2 to be found in [Carter:1972] are already the result of some special algebraic identifications. It is an interesting question whether or not these also can be discovered by machine.

It is an interesting case of a phenomenon that has occurred frequently in mathematics since computers were invented: we have here an algorithm that is locally simple, although capable of producing complicated output.

In this section, as earlier, I write $\mathfrak{e}$ for $\exp$.

We want to find coefficients $c_{k,\ell}$ such that

$$\mathfrak{e}(t\mathfrak{x}_\mu)\mathfrak{e}(sx_\lambda)=\mathfrak{e}(s\mathfrak{x}_\lambda)\mathfrak{e}(t\mathfrak{x}_\mu)\prod\nolimits_{(1,1)\leq(k,\ell)}\mathfrak{e}(c_{k,\ell}\,s^k t^\ell\mathfrak{x}_{k\lambda+\ell\mu})\,.$$

If $\lambda+\mu$ is not a root then all $\mathfrak{x}_{k\lambda+\ell\mu}$ appearing in the product vanish, and there is nothing to prove. So from now on I assume $\lambda+\mu$ to be a root.

I recall that we have assigned an order to the relevant pairs (k,ℓ) compatible with the partial ordering by the sum $k+\ell$. In particular $\lambda+\mu$ occurs in the first term of the product.

The basis of Steinberg's argument is to work with the algebraic expression

$$f(s,t) = \mathfrak{e}(-s\mathfrak{x}_\lambda)\mathfrak{e}(-t\mathfrak{x}_\mu)\mathfrak{e}(s\mathfrak{x}_\lambda)\mathfrak{e}(t\mathfrak{x}_\mu)\prod \mathfrak{e}(c_{k,\ell}\, s^k t^\ell \mathfrak{x}_{k\lambda+\ell\mu})\,.$$

It is a matrix whose enties are polynomials in s,t. We want to choose the $c_{k,\ell}$ so that $f(s,t)\equiv 1$. This will be by induction on the given linear order of the roots in the span of λ, μ. It suffices to arrange things so that the derivative of $f(s,t)$ with respect to s vanishes, since $f(0,t)$ is identically equal to I.

Let $D = s\partial/\partial s$. The dependence on s and t is somewhat illusory, and it will shorten notation to define $\mathfrak{X}_{k\lambda+\ell\mu} = s^k t^\ell \mathfrak{x}_{k\lambda+\ell\mu}$. Since

$$D\mathfrak{e}(s\mathfrak{x}) = s\mathfrak{x}\,\mathfrak{e}(s\mathfrak{x})\,,$$

we also have

$$\begin{aligned} D\mathfrak{e}(s^k t^\ell \mathfrak{x}_{k\lambda+\ell\mu}) &= k s^k t^\ell \mathfrak{x}_{k\lambda+\ell\mu}\,\mathfrak{e}(s^k t^\ell \mathfrak{x}_{k\lambda+\ell\mu}) \\ D\mathfrak{e}(c\mathfrak{X}_{k\lambda+\ell\mu}) &= ck\,\mathfrak{X}_{k\lambda+\ell\mu}\,\mathfrak{e}(\mathfrak{X}_{k\lambda+\ell\mu})\,. \end{aligned}$$

The product rule for derivatives therefore expresses $Df(s,t)$ as the sum of several products:

(***a***) $$[-\mathfrak{X}_\lambda]\cdot\mathfrak{e}(-\mathfrak{X}_\lambda)\mathfrak{e}(-\mathfrak{X}_\mu)\mathfrak{e}(\mathfrak{X}_\lambda)\mathfrak{e}(\mathfrak{X}_\mu)\prod\mathfrak{e}(c_{k,\ell}\mathfrak{X}_{k\lambda+\ell\mu})$$

(***b***) $$\mathfrak{e}(-\mathfrak{X}_\lambda)\mathfrak{e}(-\mathfrak{X}_\mu)\cdot[\mathfrak{X}_\lambda]\cdot\mathfrak{e}(\mathfrak{X}_\lambda)\mathfrak{e}(\mathfrak{X}_\mu)\prod\mathfrak{e}(c_{k,\ell}\mathfrak{X}_{k\lambda+\ell\mu})$$

(***c***) $$c_{1,1}\cdot\mathfrak{e}(-\mathfrak{X}_\lambda)\mathfrak{e}(-\mathfrak{X}_\mu)\mathfrak{e}(\mathfrak{X}_\lambda)\mathfrak{e}(\mathfrak{X}_\mu)\cdot[\mathfrak{X}_{\lambda+\mu}]\cdot\prod\mathfrak{e}(c_{k,\ell}\mathfrak{X}_{k\lambda+\ell\mu})$$

and the sum over $(k,\ell)>(1,1)$ of these products:

(***d***) $$\begin{aligned}&[kc_{k,\ell}]\cdot\mathfrak{e}(-\mathfrak{X}_\lambda)\mathfrak{e}(-\mathfrak{X}_\mu)\mathfrak{e}(\mathfrak{X}_\lambda)\mathfrak{e}(\mathfrak{X}_\mu)\\ &\quad\Big(\textstyle\prod_{(i,j)<(k,\ell)}\mathfrak{e}(c_{i,j}\mathfrak{X}_{i\lambda+j\mu})\Big)\cdot[\mathfrak{X}_{k\lambda+\ell\mu}]\cdot\Big(\textstyle\prod_{(k,\ell)\le(i,j)}\mathfrak{e}(c_{i,j}\mathfrak{X}_{i\lambda+j\mu})\Big)\end{aligned}$$

Now shift $\mathfrak{X}_\lambda$ and all $\mathfrak{X}_{k\lambda+\ell\mu}$ to the left. This can be done by applying, as often as one needs to, the basic relation (3.1) :

(4.1) $$\mathfrak{e}(c\mathfrak{X}_\alpha)\,\mathfrak{X}_\beta = \Big(\textstyle\sum_{m\ge 0} c^m M_{\alpha,\beta,m}\,\mathfrak{X}_{m\alpha+\beta}\Big)\mathfrak{e}(\mathfrak{X}_\alpha)\,.$$

This calculation is not to be undertaken lightly by hand. Even reducing (b), for which the final result is simple, is not particularly enjoyable. Doing it by machine is a bit elaborate, but with care not an extremely difficult task. At any given moment, the state of the calculation amounts to a stack and a list. The expression we are looking for is the sum of all items in both. The difference between list and stack is that the items on the stack are not in final form. They are of (a) a polynomial in s, t and the form of a product of terms $\mathfrak{x}_{\alpha,\beta}$ that come in two halves, with a single $\mathfrak{x}_{k,\ell}$ in the middle. While the stack is not empty, pop an items off it and apply (4.1) . Then put the new terms on the list if these are final, or on the stack if not.

For computation, it is convenient to index the terms by integers so that

$$f(s,t) = \prod_{i=0}^{p}\mathfrak{e}(\mathfrak{X}_{\gamma_i})\,,$$

in which each γ_i is a linear combination $k_i\lambda + \ell_i\mu$. The derivative Df at any given stage is a sum of terms

(4.2) $$P(c_{\le k}) \cdot \mathfrak{e}(c_0\mathfrak{X}_{\gamma_0}) \dots \mathfrak{e}(c_{k-1}\mathfrak{X}_{\gamma_{k-1}}) \cdot \mathfrak{X}_\gamma \cdot \mathfrak{e}(c_k\mathfrak{X}_{\gamma_k}) \dots \mathfrak{e}(c_{p-1}\mathfrak{X}_{\gamma_{p-1}}) \,,$$

for some $\gamma = i\lambda + j\mu$ which varies among the terms. At the beginning, (a) goes on a list, and (b), (c), (d) on a stack.

The top element is removed from the stack and replaced by a sum of terms, according to (4.1) . Each of these terms is put on the list if $k = 1$, or otherwise on the stack. Loop, as long as the stack is not empty.

More explicitly, the term in (4.2) is replaced by

$$\sum\nolimits_{m\ge 0} P(c_{\le k})\, c_{k-1}^m M_{\gamma_{k-1},\gamma,m} \cdot \\ \mathfrak{e}(c_0\mathfrak{X}_{\gamma_0}) \dots \mathfrak{e}(c_{k-2}\mathfrak{X}_{\gamma_{k-2}}) \cdot \mathfrak{X}_{m\gamma_{k-1}+\gamma} \cdot \mathfrak{e}(c_{k-1}\mathfrak{X}_{\gamma_{k-1}}) \dots \mathfrak{e}(c_{m-1}\mathfrak{X}_{\gamma_{m-1}}) \,.$$

How to characterize each term? By (a) a polynomial in the $c_{i,j}$, (b) the sequence of coefficients c_i, whch can be either constants or variables $c_{i,j}$, and roots γ_i, (c) the root γ and (d) its location, indexed by k, between γ_{k-1} and γ_k.

At the end we have an expression

$$\sum_{k,\ell} s^k t^\ell P(c_\bullet) \mathfrak{x}_{k\lambda+\ell\mu} f(s,t)$$

We want this to vanish. This can be arranged by an induction procedure that finds each $c_{k,\ell}$ in terms of various $M_{\alpha,\beta,m}$ and previous $c_{i,j}$.

The only terms of degree st are those with $c_{1,1}$ and $N_{\mu,\lambda}$. The conclusion is that $c_{1,1} = N_{\mu,\lambda}$.

5. Some explicit formulas

My program agrees with Carter's calculations. There are some interesting features for G_2, as there were also for Carter.

Suppose λ to be the short root, μ the long one. The machine produces the formula

(5.1) $$c_{3\lambda+2\mu} = (1/3)\, N_{\lambda,\lambda+\mu} N_{\mu,\lambda}^2 N_{\lambda+\mu,2\lambda+\mu} - (1/6)\, N_{\lambda,\lambda+\mu} N_{\mu,\lambda} N_{\mu,3\lambda+\mu} N_{\lambda,2\lambda+\mu} \,.$$

This is extremely close to the formula that Carter comes up with by hand calculation, but with some apparent differences.

Since $N_{\lambda,\mu}^2 = p_{\lambda,\mu}^2 = 1$ and $N_{\lambda,\mu} = -N_{\mu,\lambda}$, applying earlier definitions we can condense the N-products:

$$N_{\lambda,\lambda+\mu} N_{\lambda+\mu,2\lambda+\mu} = 2M_{\lambda+\mu,\lambda,2} \\ N_{\lambda,\lambda+\mu} N_{\mu,\lambda} N_{\mu,3\lambda+\mu} N_{\lambda,2\lambda+\mu} = 6M_{\lambda,\mu,3} N_{\mu,3\lambda+\mu} \,.$$

Since $N_{\mu,\lambda} = -N_{\lambda,\mu}$, the second can be written as The right hand side of (5.1) is therefore

$$-(2/3)\cdot M_{\lambda+\mu,\lambda,2} + (1/6)\cdot 6 \cdot M_{\lambda,\mu,3} N_{\mu,3\lambda+\mu} \,.$$

But a miraculous observation of Carter (Lemma 5.1 in [Carter:1972] is that these two expressions are in fact multiples of each other:

5.2. Lemma. *We have*

$$M_{\lambda,\mu,3} N_{\mu,3\lambda+\mu} = (1/3) M_{\lambda+\mu,\lambda,2} \,.$$

This makes the right hand side of (5.1) equal to $(-1/3)M_{\lambda,\mu,2}$.

In case λ is the long root and μ the short one, the machine produced

$$c_{2\lambda+3\mu} = (1/4)\, N_{\mu,\lambda+\mu} N_{\mu,\lambda}^2 N_{\lambda+\mu,\lambda+2\mu} - (1/12)\, N_{\mu,\lambda+\mu} N_{\mu,\lambda} N_{\lambda,\lambda+3\mu} N_{\mu,\lambda+2\mu}\,.$$

This time the final formula is

$$\big((1/4)(2) + (1/12)(6)\big) M_{\lambda,\mu,2} = (2/3) M_{\lambda,\mu,2}\,.$$

What happens depends on the configuration of the closure of (λ, μ). Again because every pair of roots can be transformed by an element of W into a system spanned by two simple roots, we can list all possibilities by examining the systems of rank two. There are seven:

(i)	$\{\lambda, \mu\}$
(ii)	$\{\lambda, \mu, \lambda+\mu\}$
(iiia)	$\{\lambda, \mu, \lambda+\mu, 2\lambda+\mu\}$
(iiib)	$\{\lambda, \mu, \lambda+\mu, \lambda+2\mu\}$
(iv)	$\{\lambda, \mu, \lambda+\mu, 2\lambda+\mu, \lambda+2\mu\}$
(va)	$\{\lambda, \mu, \lambda+\mu, 2\lambda+\mu, 3\lambda+\mu, 3\lambda+2\mu\}$
(vb)	$\{\lambda, \mu, \lambda+\mu, \lambda+2\mu, \lambda+3\mu, 2\lambda+3\mu\}$

Of these, case (i) is trivial, and (ii)-(iii) are relatively direct consequences of (3.2) . Case (iv) is only slightly more difficult, but I'll apply a somewhat sophisticated approach, applying a version of the Baker-Campbell-Hausdorff formula attributed to Zassenhaus. The same formula will in principle deal with (v) as well, but the result is less elegant than necessary, and for these I'll just follow [Carter:1972]. What is unsatisfying about both these approaches is that they lack uniformity. This is not true of the algorithm found in [Steinberg:1967], but this approach produces formulas more complicated than necessary. The basic problem is that there is apparently no unique form for the answer—many very different-looking formulas are equivalent. Carter's formulas in (v) are the most efficient.

I begin by displaying the final results for each closure configuration.

(i) $\{\lambda, \mu\}$:

$$\mathfrak{e}_\mu(u)\mathfrak{e}_\lambda(t) = \mathfrak{e}_\lambda(t)\mathfrak{e}_\mu(u)$$

(ii) $\{\lambda, \mu, \lambda+\mu\}$:

$$\mathfrak{e}_\mu(u)\mathfrak{e}_\lambda(t) = \mathfrak{e}_\lambda(t)\mathfrak{e}_\mu(u)\cdot\mathfrak{e}_{\lambda+\mu}(-N_{\lambda,\mu}tu)$$

(iiia) $\{\lambda, \mu, \lambda+\mu, 2\lambda+\mu\}$:

$$\mathfrak{e}_\mu(u)\mathfrak{e}_\lambda(t) = \mathfrak{e}_\lambda(t)\mathfrak{e}_\mu(u)\cdot\mathfrak{e}_{\lambda+\mu}(-N_{\lambda,\mu}tu)\mathfrak{e}_{2\lambda+\mu}(M_{\lambda,\mu,2}t^2u)$$

(iiib) $\{\lambda, \mu, \lambda+\mu, \lambda+2\mu\}$:

$$\mathfrak{e}_\mu(u)\mathfrak{e}_\lambda(t) = \mathfrak{e}_\lambda(t)\mathfrak{e}_\mu(u)\cdot\mathfrak{e}_{\lambda+\mu}(-N_{\lambda,\mu}tu)\mathfrak{e}_{\lambda+2\mu}(-M_{\mu,\lambda,2}tu^2)$$

(iv) $\{\lambda, \mu, \lambda+\mu, 2\lambda+\mu, \lambda+2\mu\}$:

$$\mathfrak{e}_\mu(u)\mathfrak{e}_\lambda(t) = \mathfrak{e}_\lambda(t)\mathfrak{e}_\mu(u)\cdot\mathfrak{e}_{\lambda+\mu}(-N_{\lambda,\mu}tu)\mathfrak{e}_{2\lambda+\mu}(M_{\lambda,\mu,2}t^2u)\mathfrak{e}_{\lambda+2\mu}(-M_{\mu,\lambda,2}tu^2)$$

(va) $\{\lambda, \mu, \lambda+\mu, 2\lambda+\mu, 3\lambda+\mu, 3\lambda+2\mu\}$:

$$\begin{aligned}\mathfrak{e}_\mu(u)\mathfrak{e}_\lambda(t) = \mathfrak{e}_\lambda(t)\mathfrak{e}_\mu(u) \\ \cdot \mathfrak{e}_{\lambda+\mu}(-N_{\lambda,\mu}tu)\mathfrak{e}_{2\lambda+\mu}(M_{\lambda,\mu,2}t^2u)\mathfrak{e}_{3\lambda+\mu}(-M_{\lambda,\mu,3}t^3u) \\ \cdot \mathfrak{e}_{3\lambda+2\mu}(-(1/3)M_{\lambda+\mu,\lambda,2}t^3u^2)\end{aligned}$$

(vb) $\{\lambda, \mu, \lambda+\mu, \lambda+2\mu, \lambda+3\mu, 2\lambda+3\mu\}$:

$$\begin{aligned}\mathfrak{e}_\mu(u)\mathfrak{e}_\lambda(t) = \mathfrak{e}_\lambda(t)\mathfrak{e}_\mu(u) \\ \cdot \mathfrak{e}_{\lambda+\mu}(-N_{\lambda,\mu}tu)\mathfrak{e}_{\lambda+2\mu}(-M_{\mu,\lambda,2}tu^2)\mathfrak{e}_{\lambda+3\mu}(-M_{\mu,\lambda,3}tu^3) \\ \cdot \mathfrak{e}_{2\lambda+3\mu}(-(2/3)M_{\lambda+\mu,\mu,2}t^2u^3)\end{aligned}$$

The final result might be at first a bit surprising. It turns out that the coefficient $c_{k,\ell}^{\lambda,\mu}$ in some sense depends only on k and ℓ. This phenomenon is a consequence of Steinberg's algorithm and an induction argument.

5.3. Theorem. (Chevalley) *We have*

$$c_{k,\ell}^{\lambda,\mu} = \begin{cases} (-1)^k M_{\lambda,\mu,k} & \text{for } \ell = 1 \\ (-1)^{1+\ell} M_{\mu,\lambda,\ell} & \text{for } k = 1 \\ -(1/3)\, M_{\lambda+\mu,\lambda,2} & \text{for } k = 3,\ \ell = 2 \\ -(2/3)\, M_{\lambda+\mu,\mu,2} & \text{for } k = 2,\ \ell = 3. \end{cases}$$

6. References

1. Armand Borel and Dan Mostow (editors), **Algebraic groups and discontinuous subgroups**, *Proceedings of Symposia in Pure Mathematics* **IX**, American Mathematical Society, 1966.

2. Roger Carter, **Simple groups of Lie type**, Wiley, 1972.

3. Bill Casselman, 'A simple way to compute structure constants of semi-simple Lie algebras', preprint, 2018. Available at

`https://www.math.ubc.ca/~cass/research/pdf/KottwitzConstants.pdf`

4. Arjeh Cohen, Scott Murray, and Don Taylor, 'Computing in groups of Lie type', *Mathematics of Computation* **73** (2004), 1477–1498.

5. Gerhard Hochschild, 'Algebraic groups and Hopf algebras' *Illinois Journal of Mathematics* **14** (1970), 52–65.

6. Bertram Kostant, 'Groups over $\mathbb{Z}$', pp. 90–98 in [Borel-Mostow:1966].

7. Jun Morita, 'Commutator relations in Kac-Moody groups', *Proceedings of the Japanese Academy of Sciences* **63** (1987), 21–22.

8. C. Quesne, 'Disentangling q-exponentials: a general approach', *International Journal of Theopretical Physics* **43** (2004), 545–559.

9. Robert Steinberg, **Lectures on Chevalley groups**, Yale University preprint, 1967.

This was republished by the American Mathematical Society in 2017, but a scan of the original (made with Steinberg's permission) can be found at

`https://www.math.ubc.ca/~cass/research/books.html`

It is unfortunately quite large (a `.zip` file of about 128 MB).