Computing quantum knot invariants.

SCOTT MORRISON

Email:

URL:

Abstract

AMS Classification ;

Keywords

Contents

1 Introduction

What are the Reshetikhin-Turaev invariants **[?]** of links coming from quantum groups? For each quantum group $U_q(\mathfrak{g})$ (by which we mean the quantised universal enveloping algebra of a complex simple Lie algebra $\mathfrak{g}$, see below), we have a function

$$(\text{framed links, with components labelled by irreps of } U_q(\mathfrak{g})) \to \mathbb{Z}[q, q^{-1}].$$

In this paper, I describe how one computes these invariants. In particular, I'll tell you just enough mathematics for the definition, but much more importantly, I'll tell you how to *actually* compute them, by showing you how to use a Mathematicapackage called `QuantumGroups‘`.

In fact, the package does much more than just compute quantum knot invariants. Subject to quite restrictive practical limitations,[1] the `QuantumGroups‘` packagecan

- Calculate dimensions of weight spaces and invariant spaces of tensor products of arbitrary highest weight representations, using a combinatorial model.
- Produce matrices representing the action of the generators of the quantum group $U_q(\mathfrak{g})$ on an arbitrary highest weight representation.
- Calculate bases for the invariants spaces inside tensor products of representations, or bases for intertwining maps between two such tensor products.
- Calculate the action of the universal R-matrix on pairs of representations.

By the end, you'll understand how to answer questions like:

> What is the invariant of the knot 8_{19} **TODO: picture!**, labelled by the 14 dimensional irrep of G_2?

(For the really impatient, one way is to download the `KnotTheory‘` Mathematicapackage from `http://katlas.org/`, and enter[2] the following in Mathematica:

In[1]:=`<<KnotTheory‘`

Loading KnotTheory‘ version of January 18, 2008, 18:17:28.7446.
Read more at `http://katlas.org/wiki/KnotTheory`.

In[2]:=`V = Irrep[G2][{0,1}]; K = Knot[8, 19];`

`QuantumKnotInvariant[G2, V][K][q]` **TODO: check this works!**

Out[2]=`???`

2 What's already done?

The Reshetikhin-Turaev invariants have been around for quite a while, but there hasn't been a significant tabulation of calculations, or a general purpose program to compute them. In this section I'll summarise what's already known. I'll concentrate on mentioning general purpose programs, which work for arbitrary links (or perhaps just knots). There's certainly more to say for many particular families of links.

The Jones polynomial [**?**] is the first interesting special case, when $\mathfrak{g} = 2$, and each component of the link is labelled with the two dimensional representation. Of course programs to compute this abound [**?**], as do tabulations of the invariants [**?**]. From the Jones polynomial, we can generalise in two directions:

[1] My code is inefficient, the algorithms are slow, and the computations are difficult!

[2] Don't type 'In[1]:='; Mathematicawill add this itself. See §?? for more details.

(1) Labelling the link with other irreps of $U_q(\mathfrak{sl}_2)$. When all the labels are the $n+1$ dimensional irrep, this is called the n-th coloured Jones polynomial of the link.

(2) Using the quantum group $U_q(\mathfrak{sl}_n)$, and labelling each component by the standard n dimensional irrep.

Again, there are many programs available which calculate both of these invariants, and many tabulations. It's a little unusual to see direct discussion of the invariant coming from the standard representation of $U_q(\mathfrak{sl}_n)$, however, because it turns out that these invariants, for varying n, all fit together as a two variable polynomial, the HOMFLYPT polynomial [**?**]. In particular,

$$HOMFLYPT_K(q^n, q) = RT_{U_q(\mathfrak{sl}_n),\mathbb{C}^n}(K)(q).$$

TODO: explain how to check this!

Thus to find programs or tables of these invariants, you're for the most part better off looking for the HOMFLYPT invariant. One notable exception is a program available in the `KnotTheory`` Mathematicapackage [**?**], which makes a direct calculation of the $U_q(\mathfrak{sl}_3)$ invariant, via Kuperberg's spider [**?**].

Next, the two variable Kauffman polynomial simultaneously captures all the Reshetikhin-Turaev invariants for the standard representations of the quantum groups $U_q(\mathfrak{so}(n))$, $n \geq 5$, and $U_q(\mathfrak{sp}(n))$, $n \geq 4$.

TODO: look these up, and write some formulas TODO: explain how to check these?

TODO: mention cabling formulas. what happens outside of type A?? TODO: examples in MMA

3 Installing the `QuantumGroups`` package

4 Combinatorial representation theory

... Thus the possibilities for the complex simple Lie algebra $\mathfrak{g}$ are

- $\mathfrak{sl}_{n+1}$, $n \geq 1$, also called A_n, with Dynkin diagram ???, and Cartan matrix

$$\begin{pmatrix} 2 & -1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & -1 & 2 & -1 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & 0 & \cdots & 0 & 0 & -1 & 2 \end{pmatrix},$$

- $\mathfrak{so}_{2n+1}$, $n \geq 2$, also called B_n, with Dynkin diagram ???, and Cartan matrix

$$\begin{pmatrix} 2 & -1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & -1 & 2 & -1 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & 0 & \cdots & 0 & 0 & -2 & 2 \end{pmatrix},$$

- $\mathfrak{sp}_{2n}$, $n \geq 3$, also called C_n, with Dynkin diagram ???, and Cartan matrix

$$\begin{pmatrix} 2 & -1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & -1 & 2 & -1 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 2 & -2 \\ 0 & 0 & \cdots & 0 & 0 & -1 & 2 \end{pmatrix},$$

- $\mathfrak{so}_{2n}$, $n \geq 4$, also called D_n, with Dynkin diagram ???, and Cartan matrix

$$\begin{pmatrix} 2 & -1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 2 & -1 & 0 & 0 \\ 0 & 0 & \cdots & -1 & 2 & -1 & -1 \\ 0 & 0 & \cdots & 0 & -1 & 2 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 0 & 2 \end{pmatrix},$$

  along with the 5 sporadic examples,

- E_6, E_7 and E_8, with Dynkin diagrams ???, and Cartan matrixes $\begin{pmatrix} 2 & 0 & -1 & 0 & 0 & 0 \\ 0 & 2 & 0 & -1 & 0 & 0 \\ -1 & 0 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}$,

$$\begin{pmatrix} 2 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{pmatrix} \text{ and } \begin{pmatrix} 2 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{pmatrix},$$

- F_4, with Dynkin diagram ???, and Cartan matrix $\begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -2 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$,

  and finally

- G_2, with Dynkin diagram ???, and Cartan matrix $\begin{pmatrix} 2 & -3 \\ -1 & 2 \end{pmatrix}$.

In the `QuantumGroups`` package, you can write these in either of two forms, for example $A2$ or A_2.

... and thus every representation of $U_q(\mathfrak{g})$ splits up into the simultaneous eigenspaces of the K_i. These spaces are called the 'weight spaces'. A representation V is a 'high weight' representation if there is a weight vector v so that $V = U_q(\mathfrak{g})^-(v)$.

The finite dimensional irreps of $U_q(\mathfrak{g})$ are all high weight representations, and for each dominant weight there is a single isomorphism class of such irreps. We'll thus write V_λ to denote 'the' representation with high weight λ.

The two standard problems in combinatorial representation theory are determining the weight multiplicities of an irrep (that is, determining the dimensions of the weight spaces), and determining the multiplicities of irreps inside the tensor product of two given irreps.

Both of these problems can be answered by using 'Littelmann paths', [?], and the `QuantumGroups`` packageexposes these algorithms as in the examples[3] below:

In[3]:=

```
DecomposeRepresentation[A2][Irrep[A2][{1,0}] ⊗
Irrep[A2][{0,1}]]
```

Out[3]=

```
ℂ ⊕ Irrep[A2][{1,1}]
```

In[4]:=

```
WeightMultiplicities[F4,Irrep[F4][{0,0,0,1}]]
```

Out[4]=

```
???
```

5 Explicit representations

Perhaps the most important function in the `QuantumGroups`` packageis `MatrixPresentation`, which produces explicit matrices representing the action of the quantum group generators on a representation.

In order to understand how these are produced, we need to make use of the following two results:

- Every irrep of $U_q(\mathfrak{g})$ is a subrepresentation of some tensor product of fundamental representations.

[3]Symbols such as $\otimes, \oplus$ and $\mathbb{C}$ can be entered in Mathematica by typing `<esc>c*<esc>`, `<esc>c+<esc>` and `<esc>dsC<esc>` respectively.

- Every fundamental representation is subrepresentation of some tensor product of 'minuscule representations' and 'short root representations'.

The first result is trivial; to produce the irrep with highest weight $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$, pick high weight vectors v_i in each fundamental representation V_{e_i}, and look at $U_q(\mathfrak{g})^- \left(\bigotimes_{i=1}^n v_i^{\otimes \lambda_i} \right) \subset \bigotimes_{i=1}^n V_{e_i}^{\otimes \lambda_i}$. This is an irrep, generated by a high weight vector, and so must be what we want.

We'll explain now what 'minuscule' and 'short root' representations are, and explain the easy proof of the second result. I was unable to find a reference for this statement. Although it is unsurprising, it's essential to what follows. The minuscule and short root representations can be presented completely explicitly, and we'll use these to build up everything else.

There are several equivalent characterisations of a minuscule representation. The simplest to state is ... ??? (depends on whether we've mentioned the weyl group!)

What is a short root representation ... ??? [2, §5A]

Minuscule representations must be fundamental representations, but the converse is not true. The following representations are minuscule:

Γ	minuscule representations	V_λ
A_n	all fundamental representations	$\lambda = e_i$, $1 \leq i \leq n$
B_n	the 2^n-d representation	$\lambda = e_n$
C_n	the $2n$-d defining representation	$\lambda = e_1$
D_n	the $2n$-d defining and 'spin' representations	$\lambda = e_1, e_{n-1}, e_n$
E_6	both 27-d representations	$\lambda = e_1$ or e_6
E_7	the 56-d representation	$\lambda = e_7$
E_8	none	
F_4	none	
G_2	none	

TODO: Justify this table? Every fundamental of F_4 has multiplicity. In G_2, $\{1, 0\}$ has no multiplicities, but a $\{0, 0\}$ weight space, outside the orbit of the high weight. TODO: All the others of E_6 have multiplicity...

From minuscule representations, we can build up others. In the type A case, nothing further is needed; every fundamental representation is minuscule. In the type B case, we see that

$$V_{e_n}^{\otimes 2} \cong \mathbb{C} \oplus (\bigoplus_{k=1}^{n-1} V_{e_k}) \oplus V_{2e_n},$$

and so every fundamental representation is contained in some tensor power (in particular the tensor square) of minuscule representations. **TODO: but that's not actually what we do for B_n!** In the type C case, we find that for $2 \leq k \leq n$, the tensor power $V_{e_1}^{\otimes k}$ contains one copy of the fundamental representation V_{e_k}. In the type D case, it's a little complicated! ???

Finally, of the exceptional groups, we clearly can't get anywhere at all using minuscule representations for E_8, F_4 or G_2.

In E_6, we find that $V_{e_1}^{\otimes 2}$ contains a copy of V_{e_3}, $V_{e_1} \otimes V_{e_6}$ contains a copy of V_{e_2}, and $V_{e_6}^{\otimes 2}$ contains a copy of V_{e_5}. That gets us almost all the way there — happily, then, we find a copy of V_{e_4} inside $V_{e_1} \otimes V_{e_3}$, and hence inside $V_{e_1}^{\otimes 3}$. These observations show that every representation of E_6 can be found inside tensor products of the minuscule representations.

In E_7, we have

$$\begin{aligned}
V_{e_7}^{\otimes 2} &\cong \mathbb{C} \oplus V_{e_1} \oplus V_{e_6} \oplus V_{2e_7} \\
&\supset V_{e_1}, V_{e_6} \\
V_{e_1}^{\otimes 2} &\cong \mathbb{C} \oplus V_{e_1} \oplus V_{e_3} \oplus V_{e_6} \oplus V_{2e_1} \\
&\supset V_{e_3} \\
V_{e_1} \otimes V_{e_7} &\cong V_{e_2} \oplus V_{e_7} \oplus V_{e_1+e_7} \\
&\supset V_{e_2} \\
V_{e_1} \otimes V_{e_2} &\cong V_{e_2} \oplus V_{e_5} \oplus V_{e_7} \oplus V_{e_1+e_2} \oplus V_{e_1+e_7} \\
&\supset V_{e_5} \\
V_{e_1} \otimes V_{e_3} &\cong V_{e_1} \oplus V_{e_3} \oplus V_{e_4} \oplus V_{e_6} \oplus V_{e_2+e_7} \oplus V_{e_1+e_3} \oplus V_{e_1+e_6} \oplus V_{2e_1} \\
&\supset V_{e_4},
\end{aligned}$$

and again, we find everything inside tensor products of minuscules, and in particular in a tensor power of the unique minuscule representation!

The following representations are short root representations ...

This is how we find every other fundamental representation inside tensor products of these ...

TODO: Note somewhere in here that although decompose $V \otimes W$ gives the same answers as decomposing $W \otimes V$, the two calculations might take dramatically different amounts of time, because of how the Littelmann path algorithm works. In particular, we need to know the entire weight multiplicities of the second factor, but nothing except the highest weight of the first. In E_8, we have

$$\begin{aligned}
V_{e_8}^{\otimes 2} &\cong \mathbb{C} \oplus V_{e_1} \oplus V_{e_7} \oplus V_{e_8} \oplus V_{2e_8} \\
&\supset V_{e_1} \oplus V_{e_7} \\
V_{e_1} \otimes V_{e_8} &\cong V_{e_1} \oplus \cong V_{e_2} \oplus \cong V_{e_7} \oplus \cong V_{e_8} \oplus \cong V_{e_1+e_8} \\
&\supset V_{e_2} \\
V_{e_1}^{\otimes 2} &\cong \mathbb{C} \oplus V_{e_1} \oplus V_{e_2} \oplus V_{e_3} \oplus V_{e_6} \oplus V_{e_7} \oplus V_{e_8} \oplus V_{2e_1} \oplus V_{e_1+e_8} \oplus V_{2e_8} \\
&\supset \oplus V_{e_2} \oplus V_{e_3} \oplus V_{e_6} \\
V_{e_2} \otimes V_{e_1} &\cong V_{e_1} \oplus V_{e_2} \oplus V_{e_3} \oplus V_{e_5} \oplus V_{e_6} \oplus V_{e_7} \oplus V_{e_8} \oplus V_{2e_1} \oplus V_{2e_8} \oplus V_{e_1+e_2} \oplus V_{e_1+e_7} \oplus 2V_{e_1+e_8} \oplus V_{e_2+e_8} \oplus V_{e_7+e_8} \\
&\supset V_{e_5} \\
V_{e_3} \otimes V_{e_1} &\cong V_{e_1} \oplus V_{e_2} \oplus 2V_{e_3} \oplus V_{e_4} \oplus V_{e_5} \oplus V_{e_6} \oplus V_{e_7} \oplus V_{2e_1} \oplus 2V_{e_1+e_2} \oplus V_{e_1+e_3} \oplus V_{e_1+e_6} \oplus 2V_{e_1+e_7} \oplus 2V_{e_1+e_8} \oplus V \\
&\supset V_{e_4}
\end{aligned}$$

In F_4,

$$\begin{aligned}
&V_{e_4}^{\otimes 2} \\
&V_{e_1}^{\otimes 2}
\end{aligned}$$

and finally in G_2

$$V_{e_1}^{\otimes 2} = \mathbb{C} \oplus V_{e_2} \oplus V_{2e_1} ???$$

Here is what minuscule representations look like

Here is what short root representations look like, cf Jantzen.

It is invoked as `MatrixPresentation`$[\Gamma][Z][V, \lambda, \beta]$. Here

Definition 5.1

Γ *is the Cartan type, see §**??**.*

Z *is a generator of the quantum group Γ, that is $X_i^\pm$ or K_i, for $1 \leq i \leq \mathrm{rank}\,(\Gamma)$. Compositions of generators, in the notation of **??**, and linear combinations, are also allowed. (Linear combinations must be homogeneous with respect to the weight grading.)*

V *is a representation, in the notation of §**??**.*

λ *is a weight, in the notation of §**??**; that is, a vector of integers, giving the weight as a linear combination of fundamental weights.*

β *is a symbol specifying the basis to use. Possible options are described in §**??**, but nearly always you'll use* `FundamentalBasis`*.*

5.1 Bases

The function `MatrixPresentation` takes an argument specifying the desired basis. In the current implementation, there is only one useful value – the symbol `FundamentalBasis`. While we give a description of how this basis is recursively defined below, essentially it depends on many minor details of the implementation. One should not depend on any particular properties of this basis!

A hypothetical future versions of the `QuantumGroups`` packagemight allow the use of the symbols `GelfandTsetlinBasis` and `CanonicalBasis`, with the obvious results.[4] Code implementing Gelfand-Testlin bases exists, but is not currently part of the package. Anyone interested in adding support for canonical bases should certainly contact me!

6 Invariant vectors and intertwiners

7 R-matrices and quantum knot invariants

7.1 Action of the Coxeter braid group on the quantum group

Next, we need to make use of the 'Coxeter braid group' associated to our quantum group, and the action of this braid group on the quantum group itself. This is

[4]Gelfand-Tsetlin bases are only projectively canonical.

the quantum analogue of the Weyl group action on the classical universal enveloping algebra. It's important to remember, as we approach defining quantum knot invariants, that although quantum groups outside of type A have Coxeter braid groups which are not the usual Artin braid groups, it is still the usual Artin braid group which acts of tensor products of representations!

The braid group $\mathcal{B}_{\mathfrak{g}}$ associated to a complex simple Lie algebra $\mathfrak{g}$ of rank n is

$$\mathcal{B}_{\mathfrak{g}} = \left\langle T_i, 1 \leq i \leq n \;\middle|\; \begin{array}{c} T_i T_j T_i T_j \cdots = T_j T_i T_j T_i \cdots \\ \text{with } 2, 3, 4 \text{ or } 6 \text{ factors on each side,} \\ \text{when } a_{ij} a_{ji} = 0, 1, 2 \text{ or } 3 \text{ respectively.} \end{array} \right\rangle$$

It is always infinite, and collapses to the Weyl group $W_{\mathfrak{g}}$ for $\mathfrak{g}$ if we impose the additional relations $T_i^2 = 1$.

The following formulas are simply translated from [1, §8.1A][5]:

TODO: reduced powers!, d_i

$$\begin{aligned}
T_i(X_i^+) &= -X_i^- K_i \\
T_i(X_i^-) &= -K_i X_i^+ \\
T_i(K_j) &= K_j K_i^{-a_{ij}} \\
T_i(X_j^+) &= \sum_{r=0}^{-a_{ij}} (-1)^{r-a_{ij}} q^{-rd_i} (X_i^+)^{(-a_{ij}-r)} X_j^+ (X_i^+)^{(r)} \qquad \text{if } i \neq j \\
T_i(X_j^-) &= \sum_{r=0}^{-a_{ij}} (-1)^{r-a_{ij}} q^{rd_i} (X_i^-)^{(r)} X_j^- (X_i^-)^{(-a_{ij}-r)} \qquad \text{if } i \neq j
\end{aligned}$$

As [1] point out, the claim that this defines an action of $\mathcal{B}_{\mathfrak{g}}$ by algebra automorphisms can be checked explicitly. Happily, at least for a given $\mathfrak{g}$, the `QuantumGroups‘` packagereally can do this check. **TODO: see, something**.

7.2 Quantum positive roots

The 'quantum positive roots', which are elements of $U_q(\mathfrak{g})$, are now defined as the action of certain words in the Coxeter braid group on certain of the generators X_i^+. It's possible to write the longest word in the Weyl group as a minimal length product of simple reflections in several way, and we'll use 'the long word decomposition' to mean the lexicographically least one:

$$w_0 = s_{i_1} s_{i_2} \cdots s_{i_r}.$$

we then define, following [1].

[5] In [1], the authors write the action on a slightly different presentation of the quantum group, over $\mathbb{C}[[h]]$, and I'm using the corresponding action on the quantum group defined over $\mathbb{C}(q)$

$$X^{\pm}_{U_q(\mathfrak{g}),1} = X^{\pm}_{i_1}, X^{\pm}_{U_q(\mathfrak{g}),2} = T_{i_1}(X^{\pm}_{i_2}), ldots, X^{\pm}_{U_q(\mathfrak{g}),r} = T_{i_1}T_{i_2}\cdots T_{i_{r-1}}(X^{\pm}_{i_r}).$$

Using a different decomposition of the longest word gives a different set of quantum positive roots, unlike in the classical case! [1, §8.1B]

We note that the lexicographically least longest word decompositions can be summarised as follows:

A_1	1
A_2	$1, 2, 1$
A_n	$w_0(A_{n-1}), n, n-1, \ldots, 1$
B_2	$1, 2, 1, 2$
B_3	$1, 2, 1, 3, 2, 1, 3, 2, 3$
B_4	$1, 2, 1, 3, 2, 1, 4, 3, 2, 1, 4, 3, 2, 4, 3, 4$
B_n	$w_0(A_n), (n+1) - \mathrm{rev}(w_0(A_{n-1}))$, or equivalently $w_0(A_{n-1}), (n+1) - \mathrm{rev}(w_0(A_n))$
C_n	same as B_n
D_4	$1, 2, 1, 3, 2, 1, 4, 2, 1, 3, 2, 4$
D_5	$1, 2, 1, 3, 2, 1, 4, 3, 2, 1, 5, 3, 2, 1, 4, 3, 2, 5, 3, 4$
D_n	$w_0(A_{n-1}), (n, (n-2, n-3, \ldots, 1)), (n-1, (n-2, n-3, \ldots, 2)),$ $(n, (n-2, n-3, \ldots, 3)), \ldots, (n$ or $n-1, (n-2)), (n-1$ or $n)$
E_6	$1, 2, 3, 1, 4, 2, 3, 1, 4, 3, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 3, 1$
E_7	$w_0(E_6), 7, 6, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 3, 1, 7, 6, 5, 4, 2, 3, 4, 5, 6, 7$
E_8	$w_0(E_7), 8, 7, 6, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 3, 1, 7, 6, 5, 4, 2, 3, 4, 5, 6, 7,$ $8, 7, 6, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 3, 1, 7, 6, 5, 4, 2, 3, 4, 5, 6, 7, 8$
F_4	$1, 2, 1, 3, 2, 1, 3, 2, 3, 4, 3, 2, 1, 3, 2, 3, 4, 3, 2, 1, 3, 2, 3, 4$
G_2	$1, 2, 1, 2, 1, 2$

The expression for D_n, which is quite complicated, includes some extra parentheses to help you see the pattern. Whether $w_0(D_n)$ ends with an $n-1$ or an n depends on whether n is odd or even, respectively.

The `QuantumGroups‘` package can tell you these decompositions, as follows

In[5]:=`LongestWordDecomposition[D4]`

Out[5]=`{1, 2, 1, 3, 2, 1, 4, 2, 1, 3, 2, 4}`

although it's worth admitting that it's not actually calculating these from scratch. In principle it can, and will produce a list of reflection matrices representing the Weyl group elements with respect to the fundamental basis, as for example

In[6]:=`WeylGroup[A2]`

Out[6]= $$\left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}, \begin{pmatrix} -1 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & -1 \end{pmatrix}, \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \right\}$$

but in practice it's inefficient enough that, for example, finding the longest word decomposition for E_8 is completely impractical. I've included decompositions calculated by the "chevie" package in "GAP".

By default, the `QuantumGroups‘` package leaves quantum roots unevaluated, and you need to explicitly apply the function `ExpandQuantumRoots` in order to reexpress them in terms of the algebra elements $X_i^\pm$. Thus, for example, we have

In[7]:= `ExpandQuantumRoots[A2] /@` $\{X^+_{A2,1}, X^+_{A2,2}, X^+_{A2,3}\}$

Out[7]= $\left\{(X_1)^+, -(X_1)^+(X_2)^+ + q^{-1}(X_2)^+(X_1)^+, (X_2)^+\right\}$

agreeing with [1, Example 8.1.5], and the most complicated of the quantum positive roots for D_4,

In[8]:= `ExpandQuantumRoots[D4][`$X^+_{D4,7}$`]`

Out[8]= $$\begin{aligned}&-q^{-1}X_1^+X_2^+X_4^+X_2^+X_3^+ + q^{-2}X_1^+X_2^+X_4^+X_3^+X_2^+ + q^{-2}X_2^+X_1^+X_4^+X_2^+X_3^+\\ &-q^{-3}X_2^+X_1^+X_4^+X_3^+X_2^+ + X_2^+X_3^+X_1^+X_2^+X_4^+ - q^{-1}X_2^+X_3^+X_2^+X_1^+X_4^+\\ &-q^{-1}X_2^+X_3^+X_4^+X_1^+X_2^+ + q^{-2}X_2^+X_3^+X_4^+X_1^+X_1^+ - q^{-1}X_3^+X_2^+X_1^+X_2^+X_4^+\\ &+q^{-2}X_3^+X_2^+X_2^+X_1^+X_4^+ + q^{-2}X_3^+X_2^+X_4^+X_1^+X_2^+ - q^{-3}X_3^+X_2^+X_4^+X_2^+X_1^+\\ &+q^{-2}X_4^+X_1^+X_2^+X_2^+X_3^+ - q^{-3}X_4^+X_1^+X_2^+X_3^+X_2^+ - q^{-3}X_4^+X_2^+X_1^+X_2^+X_3^+\\ &\quad +q^{-4}X_4^+X_2^+X_1^+X_3^+X_2^+\end{aligned}$$

7.3 The universal R-matrix

Note that there's a mistake in Chari and Pressley here ...

7.4

References

[1] **Vyjayanthi Chari**, **Andrew Pressley**, *A guide to quantum groups*, Cambridge University Press, Cambridge (1995), `MR1358358` (preview at google books)

[2] **Jens Carsten Jantzen**, *Lectures on Quantum Groups*, volume 6 of *Graduate Studies in Mathematics*, American Mathematical Society, Providence, RI (1996), `MR1359532` (preview at google books)

This paper is available online at arXiv:`arXiv:?????`, and at `http://tqft.net/quantum_groups`.

Email:

URL: