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

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

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] `QuantumGroups‘` can

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

- 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‘` Mathematica-package 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]:=`QuantumKnotInvariant[G2,Irrep[G2][{0,1}]][Knot[8,19]][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) 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.

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

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 $U_q\left(\mathfrak{sl}_n\right)$, however, because it turns out that these invariants, for varying n, all fit together as a two variable polynomial, the HOMFLYPT polynomial [**?**]. In particular,

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

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 $U_q\left(\mathfrak{sl}_3\right)$ 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

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 ???,
- $\mathfrak{so}_{2n+1}$, $n \geq 2$, also called B_n, with Dynkin diagram ???,
- $\mathfrak{sp}_{2n}$, $n \geq 3$, also called C_n, with Dynkin diagram ???,
- $\mathfrak{so}_{2n}$, $n \geq 4$, also called D_n, with Dynkin diagram ???, along with the 5 sporadic examples,
- E_6, E_7 and E_8, with Dynkin diagrams ???,
- F_4, with Dynkin diagram ???, and finally
- G_2, with Dynkin diagram ???.

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

... and thus every representation of $U_q\left(\mathfrak{g}\right)$ 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 $V = U_q\left(\mathfrak{g}\right)^-\left(v\right)$.

The finite dimensional irreps of $U_q\left(\mathfrak{g}\right)$ 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‘` package exposes 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‘` package is `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 to results:

- Every irrep of $U_q(\mathfrak{g})$ is a subrepresentation of some tensor product of fundamental representations.
- 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 that every representation can be found inside tensor products of representations which we can present as explicitly as we can the minuscule and short root representations.

There are several equivalent characterisations of a minuscule representation. The simplest to state is ... ???

[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.

What is a short root representation ... ???

The following representations are minuscule ...

The following representations are short root representations ...

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

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!

Future versions of the `QuantumGroups‘` package may 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!

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

6 Invariant vectors and intertwiners

7 R-matrices and quantum knot invariants

References

This paper is available online at arXiv:?????, and at http://scott-morrison.org/quantum_groups.

Email:
URL: