FORDHAM UNIVERSITY
THE JESUIT UNIVERSITY OF NEW YORK

A NEW PROOF FOR THE FUSION RULES OF THE LIE ALGEBRA $\mathfrak{sl}_3$

John Wu and Dr. David Swinarski ~ April 10, 2014

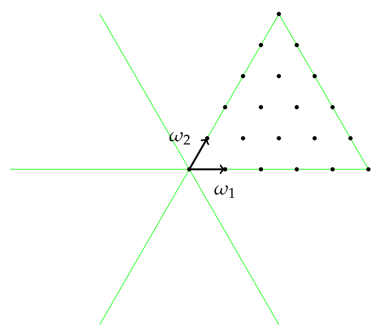
Introduction

Fusion rules in mathematical physics are a set of numbers associated to a Lie algebra. One application of the fusion rules is in computing the rank of a conformal block. Conformal blocks are a concept developed by early string theorists and have applications in mathematical physics. There is a published formula by Bégin, Mathieu, and Walton [5] that describes the fusion rules for the Lie algebra $\mathfrak{sl}_3$. My research concerns finding a new proof of this formula using the Kac-Walton algorithm.

The Lie algebra $\mathfrak{sl}_3$ and its fusion rules

Lie algebras are named after the Norwegian mathematician Sophus Lie (pronounced "Lee"). The Lie algebra $\mathfrak{sl}_3$ is the set of all $3x3$ matrices with trace zero. Below on the left is an example of such a matrix. The trace of the matrix is $a+e+i=0$. For the general definition of roots, weights, and the fundamental alcove, which are used in the Kac-Walton algorithm, see [2, §5]. As an example, the fundamental alcove of $\mathfrak{sl}_3$ at level 5 is shown below on the right.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$



For each set of three vectors in the fundamental alcove, we get an integer from the Kac-Walton algorithm described below. The set of all of these numbers is called the fusion rules.

The Kac-Walton algorithm

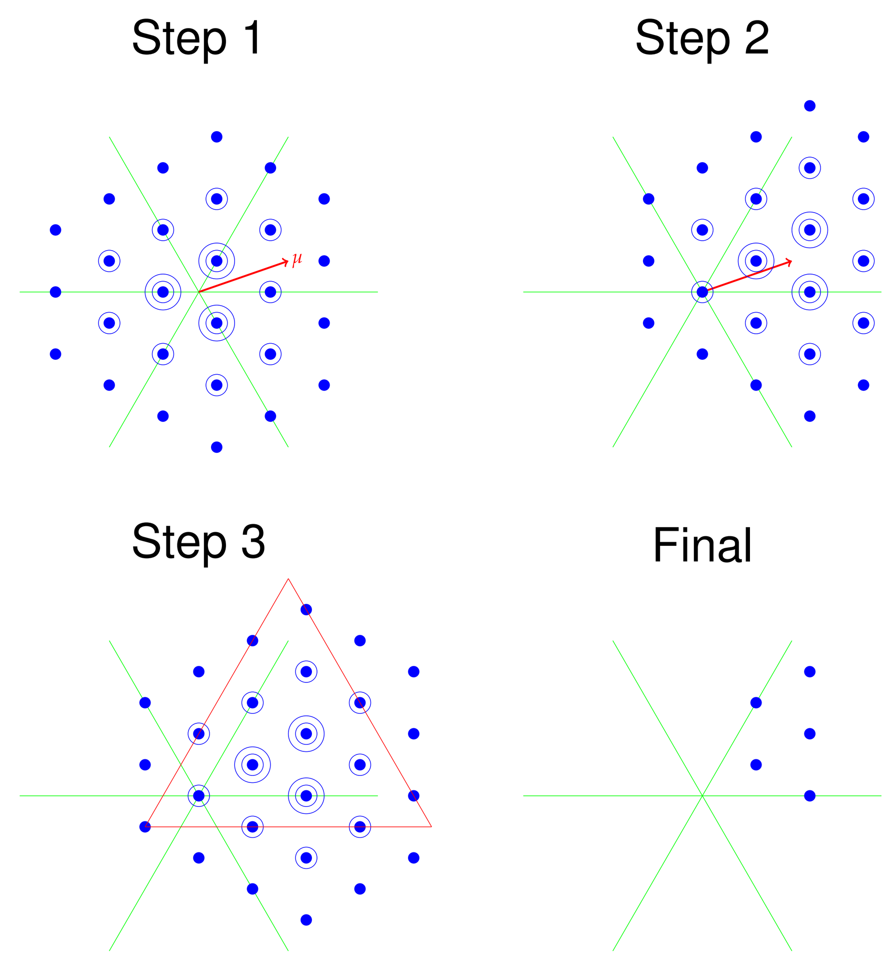
The Kac-Walton algorithm [1, §16.2.2] can be used to compute the fusion rules for any Lie algebra $\mathfrak{g}$. Let λ, μ, and ν be three vectors in the fundamental alcove of level ℓ for $\mathfrak{g}$.

Step 1. Generate the weight diagram of λ using Freudenthal's algorithm. This is an array of integers, which we designate by dots and rings.
Step 2. Translate the weight diagram by μ.
Step 3. Reflect the points outside the fundamental alcove back into the fundamental alcove.

The number of dots and rings in position ν in the resulting diagram is the fusion rule for λ, μ, and ν.

Example

This series of figures illustrates the Kac-Walton algorithm for $\lambda=(2,3)$ and $\mu=(2,1)$ at level 5. First, we generate the weight diagram for $\lambda=(2,3)$. Next, we translate by the vector $\mu=(2,1)$ (in red). The third picture illustrates the reflections across all three sides of the fundamental alcove. The fusion rules can be read from the final picture.



From this we read that

$$\begin{aligned}
&\text{the fusion rule of } (2,3),(2,1),(1,4)=1\\
&\text{the fusion rule of } (2,3),(2,1),(2,2)=1\\
&\text{the fusion rule of } (2,3),(2,1),(3,0)=1\\
&\text{the fusion rule of } (2,3),(2,1),(1,1)=1\\
&\text{the fusion rule of } (2,3),(2,1),(0,3)=1\\
&\text{and the fusion rule of} (2,1),(1,1),\nu)=0 \text{ otherwise.}
\end{aligned}$$

Formulas

The Bégin-Mathieu-Walton formula for fusion rules of the Lie algebra $\mathfrak{sl}_3$ can be written as follows: Let $\lambda=\lambda_1\omega_1+\lambda_2\omega_2=(\lambda_1,\lambda_2)$. Similarly, let $\mu=(\mu_1,\mu_2)$ and $\nu=(\nu_1,\nu_2)$. Define:

$$G(\lambda,\mu,\nu,\ell):=\begin{cases}\ell_0^{\max}-k_0^{\min}+1 & \text{if } \ell_0^{\max}-k_0^{\min}\geq -1,\\ 0 & \text{otherwise.}\end{cases}$$

where

$$\begin{aligned}
\mathcal{A}&=\frac{1}{3}(2(\lambda_1+\mu_1+\nu_2)+(\lambda_2+\mu_2+\nu_1)),\\
\mathcal{B}&=\frac{1}{3}((\lambda_1+\mu_1+\nu_2)+2(\lambda_2+\mu_2+\nu_1)),\\
k_0^{\min}&=\max\{\lambda_1+\lambda_2,\mu_1+\mu_2,\nu_1+\nu_2,\mathcal{A}-\lambda_1,\mathcal{A}-\mu_1,\mathcal{A}-\nu_2,\\
&\qquad \mathcal{B}-\lambda_2,\mathcal{B}-\mu_2,\mathcal{B}-\nu_1\},\\
\ell_0^{\max}&=\min\{\mathcal{A},\mathcal{B},\ell\}.
\end{aligned}$$

Conjecture. *If $\mathcal{A}$ and $\mathcal{B}$ are integers, then $G(\lambda,\mu,\nu,\ell)$ is the fusion rule of (λ,μ,ν). Otherwise, the fusion rule is zero.*

Let us redo the previous example using the Bégin-Mathieu-Walton formula for the nonzero fusion rules.

Let $\lambda=(2,3),\mu=(2,1),\nu=(1,4),\ell=5$. Then $G(\lambda,\mu,\nu)=1$
Let $\lambda=(2,3),\mu=(2,1),\nu=(2,2),\ell=5$. Then $G(\lambda,\mu,\nu)=1$
Let $\lambda=(2,3),\mu=(2,1),\nu=(3,0),\ell=5$. Then $G(\lambda,\mu,\nu)=1$
Let $\lambda=(2,3),\mu=(2,1),\nu=(1,1),\ell=5$. Then $G(\lambda,\mu,\nu)=1$
Let $\lambda=(2,3),\mu=(2,1),\nu=(0,3),\ell=5$. Then $G(\lambda,\mu,\nu)=1$

In order to use the Kac-Walton algorithm to give an alternative proof for the Bégin-Mathieu-Walton formula, we first need a formula for the weight diagrams in Step 1. Two FCLC students, Amy Barker and Lauren Vogelstein, found and proved a formula for the numbers that appear in weight diagrams for $\mathfrak{sl}_3$. Their formula for the multiplicity of $\nu\in WD(\lambda)$ is as follows: Let $\lambda=(\lambda_1,\lambda_2)$ and $\nu=(\nu_1,\nu_2)$ be two vectors in the fundamental alcove, and suppose $\lambda_1+2\lambda_2-\nu_1-2\nu_2$ is divisible by 3. Then the multiplicity of ν in $WD(\lambda)$ is

$$\min\left\{\frac{1}{3}(\lambda_1+2\lambda_2-\nu_1-2\nu_2)+1,\frac{1}{3}(2\lambda_1+\lambda_2-2\nu_1-\nu_2)+1,\lambda_1+1,\lambda_2+1\right\}$$

if this number is nonnegative, and zero otherwise.

Current work

We can use Barker and Vogelstein's formula for the multiplicities in the weight diagram as the input for Step 3 of the Kac-Walton Algorithm. What makes this difficult is that there is not just one expression for the multiplicity; instead, there are several expressions that must be used in different cases, and there are a large number of cases. A theoretical upper bound for the number of cases that must be checked is

$$42^{13}=1,265,437,718,438,866,624,512,$$

but in fact, we suspect that the true of number of cases that must be checked is much smaller than this. Our strategy is to write a computer program that will check each case. We use two software packages `Macaulay2` and `polymake`.

One case to us is a choice of expression for each term that contributes to the Kac-Walton algorithm. We use `Macaulay2` to compute the output of the Kac-Walton algorithm in each case. Many cases can lead to the same final output. For each output expression, we gather all the cases that give this answer. We need to show that the union of the cases associated with this answer matches the output of the Bégin-Mathieu-Walton formula.

As an example, consider the expression $e+1$. For simplicity, suppose that at most one reflection is required in the Kac-Walton algorithm. Then 6 cases yield the expression $e+1$. I have the inequalities that determine each case, and each case is a 7-dimensional cone. Using `polymake`, we can analyze how these cones are arranged, and check that the union is equal to the cone associated to this expression by the Bégin-Mathieu-Walton formula.

Currently, we need to analyze each expression by hand. We will automate this analysis and extend our program to allow for more reflections, and thus obtain a new proof of the Bégin-Mathieu-Walton formula.

References

[1] Philippe Di Francesco, Pierre Mathieu, and David Sénéchal, *Conformal field theory*, Graduate Texts in Contemporary Physics, Springer-Verlag, New York, 1997.
[2] Karin Erdmann and Mark J. Wildon, *Introduction to Lie algebras*, Springer Undergraduate Mathematics Series, Springer-Verlag London Ltd., London, 2006.
[3] Ewgenij and Joswig Gawrilow Michael, `polymake`*: a framework for analyzing convex polytopes* (2012). Version 2.12.
[4] Dan Grayson and Mike Stillman, `Macaulay2`*: a software system for research in algebraic geometry* (2012). Version 1.4.
[5] Mathieu P. Bégin L. Walton M. A., $\widehat{su}(3)_k$ *Fusion Coefficients*, Modern Physics Letters **7** (1992), no. 35, 3255–3265.