PRUV Final Report

Lucas Fagan

August 2020

1 Summary

I examined the root system of GL_n in order to understand and calculate the $\underline{i}$-trails between $\Lambda_i \to w_0 s_i \Lambda_i$. Dr. Leslie, along with work from Berenstein and Zelevinsky, has developed a formula to easily calculate monomails that are used for integrals in studying Mirkovic-Vilonen polytopes in a non-archimedean context. Specifically, I automated the process of calculating the $\underline{i}-$trails and the corresponding monomials for GL_n, and then calculated some of these MV integrals in order to study the behavior of these integrals off the polytope with the idea of resonance.

2 Reduced Words and Calculating Monomials

This part of the project required a number of combinatorial steps, and my work served to automate these steps using Python code to facilitate calculation of the MV integrals of the next section. The first step is based on the following proposition, from Berenstein and Zelevinsky and simplified by Dr. Leslie:

Proposition 2.1. *Fix a fundamental weight Λ_i and a long word $\underline{i}$. Let $w_0 s_i \Lambda_i = v_i \Lambda_i$ be a* minimal representative*. The $\underline{i}$-trails from Λ_i to $w_0 s_i \Lambda_i$ are in bijection with subwords $(i_{k(1)}, \dots, i_{k(p)})$ of $\underline{i}$ where $l(v_i) = p$ and $k(1) < \cdots < k(p)$ that are reduced words for v_i^{-1}.*

Using this proposition, I wrote a function to calculate the subwords that are reduced words for v_i^{-1}. This function has three discrete sub-functions, which each accomplish a specific task in building up the subwords.

First, I wrote a sub-function to generate all of the possible long words. This is important because each long word gives me another test case, and looking at all of the long words is important for understanding the context in general. The number of long words was found to be OEIS sequence A005118, and are given by

$$a_n = \frac{\binom{n}{2}!}{\prod_{i=0}^{n-2} (2i+1)^{-i+n-1}}.$$

This was done through applying the "braid" and "switch" relations in a recursive manner until there is no longer a way to apply either to get a unique long word.

The next sub-function calculated the v_i^{-1}, which was done through a recursive process that continued to shorten $w_0 s_i \Lambda_i$ until a minimal representative was reached. This is done through four techniques: the first two are to remove simple reflections off the right side that are inverses of each other (i.e., if the same reflection appears twice, it can be cancelled), and removing simple reflections that are not i, as $s_j \Lambda_i = \Lambda_i$ when $j \neq i$, as can be shown relatively trivially. Once it cannot be shortened through these two techniques, the algorithm applies the braid and switch relation until one of them can apply. If, after applying every possible braid and switch, it cannot be shortened any more, then it must be a minimal representative, and the inverse is returned.

The third step is to use these first two steps to calculate subwords that are reduced words for v_i^{-1}, which is done by looking at the powerset of the long word (without the empty set) and checking if each subset is equivalent to v_i^{-1}, and then only taking those of minimum length (the reduced ones). The function ends by exporting all this data into a nicely formatted CSV to facilitate analysis.

The second function calculates the monomials that are used in the MV integrals. This is done according to the following formula from Dr. Leslie:

$$\mathfrak{s}_i(u) = \sum_{\pi:\Lambda_i \to w_0 s_i \Lambda_i} d_\pi \frac{b_1^{c_1(\pi)} \cdots b_N^{c_N(\pi)}}{b_1^{\langle \Lambda_i, \beta_i^i \rangle} \cdots b_N^{\langle \Lambda_i, \beta_N^i \rangle}}.$$

Just like the last function, this one contains three discrete steps. The first sub-function generates the $\underline{i}$-trails from the subword calculated in the first function, and then uses this to trivially produce the c vectors. This is important because it determines the exponents in the numerator of the sum that we are using. The second sub-function calculates the β_i, and then uses the Λ_i to calculate the denominators of the monomials. The third sub-function combines these and simplifies the fractions to produce the final monomials. This allows us to write down the actual sum without any hand calculation, which means that we can focus on the integrals. All of the aformentioned code is provided in section 4.

3 The MV Integrals

I first state some results which are important in understanding the integrals. We define

$$I(a,b) = \begin{cases} q^{a+b} \int_{\varpi^a \mathcal{O}} \psi(t) dt & : b = 0 \\ q^{a+b} \int_{\varpi^a \mathcal{O}^\times} \psi(t) dt & : b > 0 \end{cases}$$

First, there are important vanishing statements, which appear often and allow for easy simplification.

Lemma 3.1. *$I(a,b) = 0$ if $a < -1$ or if $a = -1$ and $b = 0$.*

Proof. For $a < -1$, it vanishes because one of the terms in the product that comes out of $\psi(t)$ is $\sum_{a=0}^{p-1} e^{2\pi i a/p}$, which is 0, and for $a = -1$ and $b = 0$, the integral can be rewritten as

$$\int_{\mathbb{Z}_p} \psi(t)dt + \int_{p^{-1}\mathbb{Z}_p^\times} \psi(t)dt.$$

Given that ψ on $\mathbb{Z}_p$ is 1, this is equivalent to $1 + \sum_{a=1}^{p-1} \int_{ap^{-1}+\mathbb{Z}_p} \psi(t)dt$. Using a change of variables $t = ap^{-1} + y$ for $y \in \mathbb{Z}_p$, we get $dt = dy$ and the integral equals

$$\sum_{a=1}^{p-1} e^{2\pi i a/p} \int_{\mathbb{Z}_p} dy = (-1)(1),$$

which makes $I(a, b)$ zero. $\square$

This next lemma writes the same $I(a, b)$ in a different form which often is what actually appears in the integrals.

Lemma 3.2.

$$I(s_\alpha, m_\alpha) = \begin{cases} \int_{p^{-m_\alpha}\mathbb{Z}_p^\times} \psi(p^{s_\alpha+m_\alpha} t_\alpha)dt_\alpha) & m_\alpha > 0 \\ \int_{\mathbb{Z}_p} \psi(p^{s_\alpha} t_\alpha)dt_\alpha) & m_\alpha = 0 \end{cases}.$$

Proof. Consider the two cases. First, $m_\alpha > 0$. Then

$$I(s_\alpha, m_\alpha) = q^{s_\alpha+m_\alpha} \int_{p^{s_\alpha}\mathbb{Z}_p^\times} \psi(t)dt.$$

Then through a change of variables $t = p^{s_\alpha}x$, this is equivalent to

$$q^{m_\alpha} \int_{\mathbb{Z}_p^\times} \psi(p^{s_\alpha}x)dx$$

On the other hand,

$$\int_{p^{-m_\alpha}\mathbb{Z}_p^\times} \psi(p^{s_\alpha+m_\alpha} t_\alpha)dt_\alpha = p^{m_\alpha} \int_{\mathbb{Z}_p^\times} \psi(p^{s_\alpha}x)dx,$$

through a change of variables $t = p^{-m_\alpha}x$ with $x \in \mathbb{Z}_p^\times$.

For the second case, $m_\alpha = 0$. Thus we have

$$I(s_\alpha, m_\alpha) = q^{s_\alpha} \int_{p^{s_\alpha}\mathbb{Z}_p} \psi(t)dt = \int_{\mathbb{Z}_p} \psi(p^{s_\alpha}x)dx$$

through the same $t = p^{s_\alpha}x$ change of variables. This is equivalent to the desired integral. $\square$

3.1 GL_3 example

I will demonstrate these calculations for the simplest example, GL_3. For GL_3, there are two long words, $s_1s_2s_1$ and $s_2s_1s_2$.

Consider $\underline{i} = (1,2,1)$. This induces the ordering $\alpha_2 < \alpha_1 + \alpha_2 < \alpha_1$ on the positive roots. There are two $\underline{i}$-trails from $\Lambda_1 \to w_0 s_1 \Lambda_1$ and one from $\Lambda_2 \to w_0 s_2 \Lambda_2$. We can then calculate that

$$\mathfrak{s}_1 = \frac{1}{b_2} + \frac{b_3}{b_1 b_2} \text{ and } \mathfrak{s}_2 = \frac{1}{b_3}.$$

Because there are the same number of $\underline{i}$-trails as there are positive roots, we have that the monomial change of variables to the X_k only leads to terms with X_k to the power of zero or one. This means that the construction of the g_is is trivial, giving

$$g_1 = t_2 + \frac{t_1 w_2}{w_3} \text{ and } g_2 = t_3,$$

and we get that $s_1 = \lambda_1 + m_3 - m_2 - m_1$, $s_2 = \lambda_1 - m_2$, and $s_3 = \lambda_2 - m_3$.

This gives

$$I_\lambda(m) = \int_{C^{\underline{i}}(m)} f(u)\psi\Big(p^{\lambda_1}\big(t_2 + \frac{t_1 w_2}{w_3}\big) + p^{\lambda_2} t_3\Big)du.$$

We have $f(u)du = \prod_\alpha (p^{-1}x_\alpha)^{m_\alpha} dt_\alpha$, giving

$$\prod_\alpha (p^{-1}x_\alpha)^{m_\alpha} \int \psi(p^{\lambda_1}t_2) \int \psi(p^{\lambda_2}t_3) \int \psi\Big(p^{\lambda_1}\frac{t_1 w_2}{w_3}\Big)dt.$$

Now consider

$$G(s_1, m_1) = \int \psi\Big(p^{\lambda_1}\frac{t_1 w_2}{w_3}\Big)dt_1.$$

Due to the vanishing conditions of $I(a,b)$, we have that if $s_1 < -1$, or if $s_1 = -1$ and $m_1 = 0$, then $G(s_1, m_1) = 0$.

Let $m_1 > 0$. Then

$$\int_{p^{-m_1}\mathbb{Z}_p^\times} \psi\Big(p^{\lambda_1}\frac{t_1 t_2}{t_3}\Big)dt_1 = p^{m_1}\int_{\mathbb{Z}_p^\times} \psi\Big(p^{\lambda_1 - m_1}\frac{x t_2}{t_3}\Big)dx.$$

If $s_1 \geq 0$, this integral equals

$$p^{m_1}\int_{\mathbb{Z}_p^\times} dx = p^{m_1}(1 - p^{-1}).$$

If $s_1 = -1$, we can re-write the integral as

$$p^{m_1}\sum_{a=1}^{p-1}\int_{a+p\mathbb{Z}_p} \psi\Big(p^{\lambda_1}\frac{t_2}{t_3}x\Big)dx$$

and through a change of variables $x = a + py$ we get

$$p^{m_1-1}\sum_{a=1}^{p-1}\psi\Big(p^{\lambda_1}\frac{t_2}{t_3}a\Big)\int_{\mathbb{Z}_p}\psi\Big(p^{\lambda_1+1}\frac{t_2}{t_3}y\Big)dy.$$

The integral on the right is 1, which means the sum evaluates to -1, leaving $-p^{m_1-1}$. The last case to consider is $m_\alpha = 0$ and $s_\alpha \geq 0$. We get that

$$\int_{\mathbb{Z}_p}\psi\Big(p^{\lambda_1}\frac{t_1 t_2}{t_3}\Big)dt = 1.$$

It is clear that the same can be done for the other two integrals. Thus, we can write

$$I_\lambda(m) = \prod_\alpha x_\alpha^{m_\alpha} G(s_\alpha, m_\alpha),$$

with

$$G(s_\alpha, m_\alpha) = \begin{cases} 1 - p^{-1} & m_\alpha > 0, s_\alpha \geq 0 \\ -p^{-1} & m_\alpha > 0, s_\alpha = -1 \\ 1 & m_\alpha = 0, s_\alpha \geq 0 \\ 0 & \text{otherwise} \end{cases}.$$

3.2 GL_4 degenerate example

I now demonstrate a calculation for an "interesting" example, in GL_4. Note that there are 6 positive roots, but 7 monomials, which means that the resulting integrals are nontrivial and we have nonzero contribution outside the polytope.

Consider $\underline{i} = (2, 3, 1, 2, 3, 1)$. This induces the ordering

$$\alpha_2 < \alpha_1 + \alpha_2 < \alpha_2 + \alpha_3 < \alpha_1 + \alpha_2 + \alpha_3 < \alpha_3 < \alpha_1$$

on the positive roots. There is one $\underline{i}-$trail from $\Lambda_1 \to w_0 s_1 \Lambda_1$, five from $\Lambda_2 \to w_0 s_2 \Lambda_2$, and one from $\Lambda_3 \to w_0 s_2 \Lambda_3$. We get

$$\mathfrak{s}_1 = \frac{1}{b_5} = X_5$$

$$\mathfrak{s}_2 = \frac{1}{b_4} + \frac{b_6}{b_3 b_4} + \frac{b_5}{b_2 b_4} + \frac{b_5 b_6}{b_2 b_3 b_4} + \frac{b_s b_6}{b_1 b_2 b_3} = X_4 + X_3 + \frac{X_2 X_4}{X_3} + X_2 + X_1$$

$$\mathfrak{s}_3 = \frac{1}{b_6} = X_6,$$

which gives

$$g_1 = t_5, \quad g_2 = t_4 + \frac{t_3 w_4}{w_6} + \frac{t_2 t_4}{w_5} + \frac{t_2 w_3 w_4}{w_5 w_6} + \frac{t_1 w_2 w_3}{w_5 w_6}, \quad g_3 = t_6.$$

Thus we have

$$s_1 = \lambda_2 - m_1 - m_2 - m_3 + m_5 + m_6, \quad s_2 = \lambda_2 - m_2 - m_3 - m_4 + m_5 + m_6$$

$$s_3 = \lambda_2 - m_3 - m_4 + m_6, \quad s_4 = \lambda_2 - m_4, \quad s_5 = \lambda_1 - m_5, \quad s_6 = \lambda_3 - m_6$$

$$s_2 + s_4 - s_3 = \lambda_2 - m_2 - m_4 + m_5.$$

This gives

$$I_\lambda(\mathbf{m}) = \int_{C^{\underline{i}}(m)} f(u)\, \psi\left(p^{\lambda_1} t_5 + p^{\lambda_3} t_6 + p^{\lambda_2}\left(t_4 + \frac{t_3 w_4}{w_6} + \frac{t_2 t_4}{w_5} + \frac{t_2 w_3 w_4}{w_5 w_6} + \frac{t_1 w_2 w_3}{w_5 w_6}\right)\right) du$$

which we can simplify by writing as

$$\int \psi\left(p^{\lambda_2} \frac{t_1 w_2 w_3}{w_5 w_6}\right) \int\int\int\int\int \psi\left(p^{\lambda_1} t_5 + p^{\lambda_3} t_6 + p^{\lambda_2}\left(t_4 + \frac{t_3 w_4}{w_6} + \frac{t_2 t_4}{w_5} + \frac{t_2 w_3 w_4}{w_5 w_6}\right)\right) \prod_\alpha \left(p^{-1} x_\alpha\right)^{m_\alpha} dt_\alpha$$

$$= I(s_1, m_1) J_\lambda(\mathbf{m}) \prod_\alpha \left(p^{-1} x_\alpha\right)^{m_\alpha}$$

Case 1: $s_2 + s_4 - s_3 \geq 0$

In this case we see that there is no contribution from the $t_2 t_4 / w_5$ term, and thus we can write

$$J_\lambda(\mathbf{m}) = I(s_5, m_5) I(s_6, m_6) q^{m_2 + m_3 + m_4} \int\int\int \psi\left(\varpi^{s_2} t_2 + \varpi^{s_3} t_3 + \varpi^{s_4} t_4\right) dt_2 dt_3 dt_4,$$

and we get the standard contribution. Thus now we assume $s_2 + s_4 - s_3 < 0$

Case 2: $s_2 = -1, s_4 = s_3 \geq 0$

Note that we have $s_2 + s_4 - s_3 = -1$, and further that $m_6 = m_3$. Thus for this case, consider $s_2 = \lambda_2 - m_2 - m_4 + m_5$. This demonstrates that we cannot have $m_2 = m_4 = 0$, because we cannot have that $\lambda_2 + m_5 = -1$.

We can simplify $J_\lambda(\mathbf{m})$ by writing

$$J_\lambda(\mathbf{m}) = I(s_5, m_5) I(s_6, m_6) I_\lambda(s_2, s_3, s_4; m_2, m_3, m_4)$$

and do a change of variables to write $I_\lambda(s_2, s_3, s_4; m_2, m_3, m_4)$ as

$$q^{m_2 + m_3 + m_4} \int\int\int \psi\left(\varpi^{s_2} y_2 + \varpi^{s_3} y_3 + \varpi^{s_2 + s_4 - s_3} \frac{y_2 y_4}{y_3} + \varpi^{s_4} y_4\right) dy_2 dy_3 dy_4.$$

Because we have $s_4 = s_3 \geq 0$, we have that the $\varpi^{s_3} y_3$ and $\varpi^{s_4} y_4$ do not contribute, and thus we can write this as

$$\int \psi\left(\varpi^{-1} \frac{y_2 y_4}{y_3} + \varpi^{-1} y_2\right) dy_2 dy_3 dy_4.$$

Case 2.1 $m_2 > 0, m_3 > 0, m_4 > 0$

This is the singular nonzero subcase of Case 2. Because $s_4 = s_3 \geq 0$, we have that the $\varpi^{s_3} y_3$ and $\varpi^{s_4} y_4$ do not contribute, and we can write this triple integral

$$\int_{(\mathcal{O}^\times)^3} \psi\left(\varpi^{-1}\frac{y_2 y_4}{y_3} + \varpi^{-1} y_2\right) dy_2 dy_3 dy_4,$$

and we can do a change of variables to get

$$q^{-1}\left(1 - \frac{1}{q}\right) \int_{\varpi^{-1}\mathcal{O}^\times} \int_{\mathcal{O}^\times} \psi(xy + x) dy dx.$$

Now let $z = xy$ and we get

$$q^{-2}\left(1 - \frac{1}{q}\right) \int_{\varpi^{-1}\mathcal{O}^\times} \int_{\varpi^{-1}\mathcal{O}^\times} \psi(z + x) dz dx.$$

These two integrals both evaluate to -1 and cancel, giving

$$q^{-2}\left(1 - \frac{1}{q}\right).$$

Case 2.2 $m_2 = 0, m_3 > 0, m_4 > 0$

We have

$$\int_{(\mathcal{O}^\times)^2} \int_{\mathcal{O}} \psi\left(\varpi^{-1}\frac{y_2 y_4}{y_3} + \varpi^{-1} y_2\right) dy_2 dy_3 dy_4,$$

and we can do a change of variables to get

$$q^{-1}\left(1 - \frac{1}{q}\right) \int_{\varpi^{-1}\mathcal{O}} \int_{\mathcal{O}^\times} \psi(xy + x) dy dx.$$

Now let $z = xy$ and we get

$$q^{-2}\left(1 - \frac{1}{q}\right) \int_{\varpi^{-1}\mathcal{O}} \int_{\varpi^{-1}\mathcal{O}^\times} \psi(z + x) dz dx.$$

This outer (first) integral evaluates to 0, meaning the whole case is 0.

Case 2.3 $m_2 > 0, m_3 = m_4 = 0$

This case follows almost identically to Case 2.2, and we also get 0.

Case 2.4 $m_2 > 0, m_3 = 0, m_4 > 0$

We have

$$\int_{(\mathcal{O}^\times)^2} \int_{\mathcal{O}} \psi\left(\varpi^{-1}\frac{y_2 y_4}{y_3} + \varpi^{-1} y_2\right) dy_3 dy_2 dy_4,$$

Do a change of variables to get

$$q^{-1}\left(1-\frac{1}{q}\right)\int_{\varpi^{-1}\mathcal{O}^\times}\psi(x)\int_{\mathcal{O}}\psi\left(\frac{x}{y}\right)dydx$$

which we can write as

$$q^{-1}\left(1-\frac{1}{q}\right)\int_{\varpi^{-1}\mathcal{O}^\times}\psi(x)\left(\int_{\mathcal{O}^\times}\psi\left(\frac{x}{y}\right)+\int_{\varpi\mathcal{O}}\psi\left(\frac{x}{y}\right)\right)$$

which can be written as

$$\left(1-\frac{1}{q}\right)\int_{\mathcal{O}^\times}\psi(\varpi^{-1}x)\left(\int_{\mathcal{O}^\times}\psi\left(\varpi^{-1}y\right)+\varpi\int_{\mathcal{O}}\psi\left(\varpi^{-2}y\right)\right)$$

which equals

$$q^{-2}\left(1-\frac{1}{q}\right)$$

because the second integral is 0.

Case 2.5 $m_2>0, m_3>0, m_4=0$

We have

$$\int_{(\mathcal{O}^\times)^2}\int_{\mathcal{O}}\psi\left(\varpi^{-1}\frac{y_2y_4}{y_3}+\varpi^{-1}y_2\right)dy_3dy_2dy_4,$$

and we can simplify through a change of variables to

$$q^{-2}\left(1-\frac{1}{q}\right)\int_{\mathcal{O}^\times}\psi(x)\int_{\mathcal{O}}\psi(y)dydx$$

which equals 0 because the right integral is 0.

Case 2.6 $m_2=0, m_3=0, m_4>0$

We have

$$\int_{\mathcal{O}^\times}\int_{\mathcal{O}^2}\psi\left(\varpi^{-1}\frac{y_2y_4}{y_3}+\varpi^{-1}y_2\right)dy_3dy_2dy_4,$$

Do a change of variables to get

$$q^{-1}\left(1-\frac{1}{q}\right)\int_{\varpi^{-1}\mathcal{O}}\psi(x)\int_{\mathcal{O}}\psi\left(\frac{x}{y}\right)dydx$$

which we can write as

$$q^{-1}\left(1-\frac{1}{q}\right)\int_{\varpi^{-1}\mathcal{O}}\psi(x)\left(\int_{\mathcal{O}^\times}\psi\left(\frac{x}{y}\right)+\int_{\varpi\mathcal{O}}\psi\left(\frac{x}{y}\right)\right)$$

which can be written as

$$\left(1-\frac{1}{q}\right)\int_{\mathcal{O}}\psi(\varpi^{-1}x)\left(\int_{\mathcal{O}^\times}\psi\left(\varpi^{-1}y\right)+\varpi\int_{\mathcal{O}}\psi\left(\varpi^{-2}y\right)\right)$$

which equals 0 because the first integral is 0.

Case 2 Summary

We can summarize the case as follows:

$$I_\lambda(s_2, s_3, s_4; m_2, m_3, m_4) = \begin{cases} q^{-2}\left(1 - \frac{1}{q}\right) & m_2 > 0, m_4 > 0 \\ 0 & \text{otherwise} \end{cases}$$

Case 3 $s_4 = -1, s_2 = s_3 \geq 0$

First, note that $s_2 = s_3$ implies that $m_5 = m_2$. Further, because we have $s_4 = -1 = \lambda_2 - m_4$, we must have that $m_4 > 0$.

Note that in this case, we can do the same change of variables as in case 2, and get

$$q^{m_2+m_3+m_4} \int \psi\left(\varpi^{-1}\frac{y_2 y_4}{y_3} + \varpi^{-1} y_4\right) dy_2 dy_3 dy_4.$$

Case 3.1 $m_2 > 0, m_3 > 0$

Note that this is the same as case 2.1. We get

$$q^{-2}\left(1 - \frac{1}{q}\right).$$

Case 3.2 $m_2 = 0, m_3 > 0$

It is clear that in case 3, y_4 is playing the role of y_2 from case 2. Thus, this is equivalent to case 2.5 and we get 0.

Case 3.3 $m_2 = 0, m_3 = 0$

This is equivalent to case 2.3, which equals 0.

Case 3.4 $m_2 > 0, m_3 = 0$

This is equivalent to case 2.4, which equals

$$q^{-2}\left(1 - \frac{1}{q}\right).$$

Case 3 Summary

We can see that this is almost identical to case 2, with the additional restriction that $m_4 \neq 0$, thus giving

$$I_\lambda(s_2, s_3, s_4; m_2, m_3, m_4) = \begin{cases} q^{-2}\left(1 - \frac{1}{q}\right) & m_2 > 0 \\ 0 & \text{otherwise} \end{cases}$$

Case 4 $s_4 = s_3 = s_2 = -1$

This is the final case inside the polytope. Note that we have $m_4 > 0$ by the same logic as in case 3, and further $m_5 = m_2$ and $m_6 = m_3$.

Case 4.1 $m_2 > 0, m_3 > 0$

Do a change of variables $y_2 = \varpi^{-s_2}x$, $y_3 = \varpi^{-s_3}z$, $y_4 = \varpi^{-s_4}y$, leaving us with

$$q^{-3}\int_{\varpi^{s_2}\mathcal{O}^\times}\int_{\varpi^{s_3}\mathcal{O}^\times}\int_{\varpi^{s_4}\mathcal{O}^\times}\psi\left(x+y+\frac{xy}{z}+z\right)dzdydx.$$

Now again change variables by $y = wz$, giving

$$q^{-2}\int_{\varpi^{s_2}\mathcal{O}^\times}\int_{\varpi^{s_3}\mathcal{O}^\times}\int_{\mathcal{O}^\times}\psi\left(x+z+xw+wz\right)dwdzdx.$$

We can write this as

$$q^{-2}\int_{\varpi^{s_2}\mathcal{O}^\times}\int_{\varpi^{s_3}\mathcal{O}^\times}\psi(x+z)\int_{\mathcal{O}^\times}\psi\left(w(x+z)\right)dwdzdx.$$

Now let $w = a + \varpi v$, which gives

$$q^{-3}\int_{\varpi^{s_2}\mathcal{O}^\times}\int_{\varpi^{s_3}\mathcal{O}^\times}\psi(x+z)\sum_{a=1}^{\varpi-1}\psi\left(a(x+z)\right)\int_{\mathcal{O}}\psi\left(\varpi v(x+y)\right)dvdzdx.$$

We can again change variables to get

$$q^{-1}\int_{\mathcal{O}^\times}\int_{\mathcal{O}^\times}\psi(\varpi^{s_2}x+\varpi^{s_3}z)\sum_{a=1}^{\varpi-1}\psi\left(a\varpi^{-1}(x+z)\right)\int_{\mathcal{O}}\psi\left(v(x+z)\right)dvdzdx.$$

Let $x = x'z$, which gives

$$q^{-1}\int_{\mathcal{O}^\times}\int_{\mathcal{O}^\times}\psi(\varpi^{-1}z(x'+1))\sum_{a=1}^{\varpi-1}\psi\left(a\varpi^{-1}z(x'+1)\right)\int_{\mathcal{O}}\psi\left(vy(x'+1)\right)dvdzdx'.$$

Let $x' = b + \varpi x''$. We can write

$$q^{-2}\sum_{b=1}^{\varpi-1}\int_{\mathcal{O}}\int_{\mathcal{O}^\times}\psi(\varpi^{-1}z(b+\varpi x''+1))\sum_{a=1}^{\varpi-1}\psi\left(a\varpi^{-1}z(b+\varpi x''+1)\right)\int_{\mathcal{O}}\psi\left(vz(b+\varpi x''+1)\right)dvdzdx''.$$

In the last integral, the $\varpi x''$ term does not contribute, and it becomes clear that the integral (and thus the whole expression) is 0 for $b \neq -1$ [write out steps here]. Thus let $b = -1$. We get

$$q^{-1}\int_{\mathcal{O}}\int_{\mathcal{O}^\times}\psi(zx'')\sum_{a=1}^{\varpi-1}\psi(azx'')dzdx''.$$

This sum evaluates to $\varpi - 1$ because each term is 1, and we get

$$q^{-1}(q-1)\int_{\mathcal{O}}\int_{\mathcal{O}^\times}\psi(zx'')dydx'',$$

and through a change of variables we get

$$\begin{aligned} q^{-1}(q-1)\left(1-\frac{1}{q}\right)\int_{\mathcal{O}}\psi(x'')dx'' &= q^{-1}(q-1)\left(1-\frac{1}{q}\right) \\ &= \left(1-\frac{1}{q}\right)^2 \end{aligned}$$

Case 4.2 $m_2 = 0, m_3 > 0$

In this case we get

$$q^{-3}\int_{\varpi^{s_2}\mathcal{O}}\int_{\varpi^{s_3}\mathcal{O}^\times}\int_{\varpi^{s_4}\mathcal{O}^\times}\psi\left(x+y+\frac{xy}{z}+z\right)dydzdx.$$

and we similarly change variables to get

$$q^{-2}\int_{\varpi^{s_2}\mathcal{O}}\int_{\varpi^{s_3}\mathcal{O}^\times}\int_{\mathcal{O}^\times}\psi\left(x+z+xw+wz\right)dwdzdx.$$

Now let $w = a + \varpi v$, which gives

$$q^{-3}\int_{\varpi^{s_2}\mathcal{O}}\int_{\varpi^{s_3}\mathcal{O}^\times}\psi(x+z)\sum_{a=1}^{\varpi-1}\psi\left(a(x+z)\right)\int_{\mathcal{O}}\psi\left(\varpi v(x+y)\right)dvdzdx.$$

We can again change variables to get

$$q^{-1}\int_{\mathcal{O}}\int_{\mathcal{O}^\times}\psi(\varpi^{s_2}x+\varpi^{s_3}z)\sum_{a=1}^{\varpi-1}\psi\left(a\varpi^{-1}(x+z)\right)\int_{\mathcal{O}}\psi\left(v(x+z)\right)dvdzdx.$$

Let $x = x'z$, which gives

$$q^{-1}\int_{\mathcal{O}}\int_{\mathcal{O}^\times}\psi(\varpi^{-1}z(x'+1))\sum_{a=1}^{\varpi-1}\psi\left(a\varpi^{-1}z(x'+1)\right)\int_{\mathcal{O}}\psi\left(vy(x'+1)\right)dvdzdx'.$$

Let $x' = b + \varpi x''$. We can write

$$q^{-2}\sum_{b=0}^{\varpi-1}\int_{\mathcal{O}}\int_{\mathcal{O}^\times}\psi(\varpi^{-1}z(b+\varpi x''+1))\sum_{a=1}^{\varpi-1}\psi\left(a\varpi^{-1}z(b+\varpi x''+1)\right)\int_{\mathcal{O}}\psi\left(vz(b+\varpi x''+1)\right)dvdzdx''.$$

Case 5 $s_4 = s_3 = s_2 = -k$ **for** $k > 1$

In this case, we also have $m_4 > 0$, and furthermore that $m_5 = m_2$ and $m_6 = m_3$.

Case 5.1 $m_2 > 0, m_3 > 0$

Do a change of variables $y_2 = \varpi^{-s_2}x$, $y_3 = \varpi^{-s_3}z$, $y_4 = \varpi^{-s_4}y$, leaving us with

$$q^{-3k} \int_{\varpi^{s_2}\mathcal{O}^\times} \int_{\varpi^{s_3}\mathcal{O}^\times} \int_{\varpi^{s_4}\mathcal{O}^\times} \psi\left(x + y + \frac{xy}{z} + z\right) dydzdx.$$

Now again change variables by $y = wz$, giving

$$q^{-2k} \int_{\varpi^{s_2}\mathcal{O}^\times} \int_{\varpi^{s_3}\mathcal{O}^\times} \int_{\mathcal{O}^\times} \psi\left(x + z + xw + wz\right) dwdzdx.$$

We can write this as

$$q^{-2k} \int_{\varpi^{s_2}\mathcal{O}^\times} \int_{\varpi^{s_3}\mathcal{O}^\times} \psi(x+z) \int_{\mathcal{O}^\times} \psi\left(w(x+z)\right) dwdzdx$$

or

$$\int_{\mathcal{O}^\times} \int_{\mathcal{O}^\times} \psi\left(\varpi^{-k}(x+z)\right) \int_{\mathcal{O}^\times} \psi\left(\varpi^{-k}w(x+z)\right) dwdzdx$$

It is clear in this case that we care about the valuation of $x + z$. Thus, let $t = x + z$. We note that $|t| < |x| = |z| = 1$. Thus we can write this as

$$\left(1 - \frac{1}{q}\right) \sum_{\ell=0}^{\infty} \int_{|t|=\varpi^{-\ell}} \psi(\varpi^{-k}t) \int_{\mathcal{O}^\times} \psi\left(\varpi^{-k}wt\right) dwdt$$

and do a change of variables $t = \varpi^{\ell}t$ to get

$$\left(1 - \frac{1}{q}\right) \sum_{\ell=0}^{\infty} q^{-\ell} \int_{\mathcal{O}^\times} \psi(\varpi^{\ell-k}t) \int_{\mathcal{O}^\times} \psi\left(\varpi^{\ell-k}wt\right) dwdt.$$

Recognize this as

$$\left(1 - \frac{1}{q}\right) \sum_{\ell=0}^{\infty} q^{-\ell} I(\ell - k, 1)^2$$

and by Lemma 1.1 we have that it equals 0 if $\ell < k - 1$. Furthermore, when $\ell = k - 1$, we have that $I(\ell - k, 1)$ equals $-q^{-1}$, and when $\ell \geq k$, we have that $I(\ell - k, 1)$ equals $1 - q^{-1}$. Thus we can write this as

$$\left(1 - \frac{1}{q}\right) \left(q^{-k-1} + (1 - q^{-1})^2 \sum_{\ell=k}^{\infty} q^{-\ell}\right)$$

and this sum is a geometric series, giving

$$\left(1 - \frac{1}{q}\right) \left(q^{-k-1} + (1 - q^{-1})^2 \frac{q^{-k}}{1 - q^{-1}}\right)$$
$$= \left(1 - \frac{1}{q}\right) q^{-k}$$

Case 5.2 $m_2 = 0, m_3 > 0$

We start the same way as in Case 5.1, with the only difference being that we are integrating x over $\mathcal{O}$ instead of $\mathcal{O}$. We get

$$\int_{\mathcal{O}} \int_{\mathcal{O}^\times} \psi\left(\varpi^{-k}(x+z)\right) \int_{\mathcal{O}^\times} \psi\left(\varpi^{-k} w(x+z)\right) dwdzdx$$

We can write this as

$$\int_{\mathcal{O}^\times} \int_{\mathcal{O}^\times} \psi\left(\varpi^{-k}(x+z)\right) \int_{\mathcal{O}^\times} \psi\left(\varpi^{-k} w(x+z)\right) dwdzdx$$
$$+ \int_{\varpi\mathcal{O}} \int_{\mathcal{O}^\times} \psi\left(\varpi^{-k}(x+z)\right) \int_{\mathcal{O}^\times} \psi\left(\varpi^{-k} w(x+z)\right) dwdzdx$$

The first integral is equivalent to Case 5.1, and in the second integral we can a change of variables $t = x + z$, and we know that t has valuation 1. Thus we can recognize it as

$$q^{-1} I(-k, 1)^2$$

and we are assuming that $k > 1$, which means it is 0. Thus it evaluates to the same as in case 5.1.

Case 5.3 $m_2 > 0, m_3 = 0$

Do a change of variables $y_2 = \varpi^{-s_2} x$, $y_3 = \varpi^{-s_3} z$, $y_4 = \varpi^{-s_4} y$, and expand to get

$$\sum_{j=0}^{\infty} q^{-3k} \int_{\varpi^{s_2}\mathcal{O}^\times} \int_{\varpi^{s_3+j}\mathcal{O}^\times} \int_{\varpi^{s_4}\mathcal{O}^\times} \psi\left(x + y + \frac{xy}{z} + z\right) dydzdx.$$

We get

$$\sum_{j=0}^{\infty} q^{-j} \int_{\mathcal{O}^\times} \int_{\mathcal{O}^\times} \psi\left(\varpi^{-k}(x+z)\right) \int_{\mathcal{O}^\times} \psi\left(\varpi^{-j-k} w(x+z)\right) dwdzdx$$

And then using the same logic as in Case 5.1, we get

$$\left(1 - \frac{1}{q}\right) \sum_{j=0}^{\infty} q^{-j} \sum_{\ell=0}^{\infty} q^{-\ell} I(\ell - k, 1) I(-j + \ell - k, 1)$$

which is 0 when $\ell < k - 1 + j$, $-q^{-1}$ when $\ell = k - 1 + j$, and $1 - q^{-1}$ when $\ell \geq k + j$. Thus we can write

$$\left(1 - \frac{1}{q}\right) q^{-k} + \left(1 - \frac{1}{q}\right) \sum_{j=1}^{\infty} q^{-j} \left(-q^{-k-j}\left(1 - q^{-1}\right) + \sum_{\ell=k+j}^{\infty} q^{-\ell}(1 - q^{-1})^2 \right)$$
$$= \left(1 - \frac{1}{q}\right) q^{-k}$$

Case 5.4 $m_2 = 0, m_3 = 0$

It follows that this is equivalent to case 5.3.

Case 6 $s_2 = s_3 = -1, s_4 = -k$ **for** $k > 1$

Note that we have $m_4 > 0$ by the same logic as in case 5, and further $m_5 = m_2$.

Case 6.1 $m_2 > 0, m_3 > 0$

Do a change of variables $y_2 = \varpi^{-s_2}x$, $y_3 = \varpi^{-s_3}z$, $y_4 = \varpi^{-s_4}y$, leaving us with

$$q^{-2-k}\int_{\varpi^{s_2}\mathcal{O}^\times}\int_{\varpi^{s_3}\mathcal{O}^\times}\int_{\varpi^{s_4}\mathcal{O}^\times}\psi\left(x+y+\frac{xy}{z}+z\right)dydzdx.$$

Now again change variables by $y = wz$, giving

$$q^{-1-k}\int_{\varpi^{s_2}\mathcal{O}^\times}\int_{\varpi^{s_3}\mathcal{O}^\times}\int_{\varpi^{-k+1}\mathcal{O}^\times}\psi\left(x+z+xw+wz\right)dwdzdx.$$

We can write this as

$$q^{-2k}\int_{\varpi^{s_2}\mathcal{O}^\times}\int_{\varpi^{s_3}\mathcal{O}^\times}\psi(x+z)\int_{\mathcal{O}^\times}\psi\left(w(x+z)\right)dwdzdx$$

or

$$\int_{\mathcal{O}^\times}\int_{\mathcal{O}^\times}\psi\left(\varpi^{-1}(x+z)\right)\int_{\mathcal{O}^\times}\psi\left(\varpi^{-k}w(x+z)\right)dwdzdx$$

It is clear in this case that we care about the valuation of $x + z$. Thus, let $t = x + z$. We note that $|t| < |x| = |z| = 1$. Thus we can write this as

$$\left(1-\frac{1}{q}\right)\sum_{\ell=0}^{\infty}\int_{|t|=\varpi^{-\ell}}\psi(\varpi^{-1}t)\int_{\mathcal{O}^\times}\psi\left(\varpi^{-k}wt\right)dwdt$$

which is equivalent to

$$\left(1-\frac{1}{q}\right)\sum_{\ell=0}^{\infty}q^{-\ell}I(\ell-1,1)I(\ell-k,1)$$

or

$$\left(1-\frac{1}{q}\right)\left(-q^{-k}(1-q^{-1})+(1-q^{-1})^2\sum_{\ell=k}^{\infty}q^{-\ell}\right)$$

which equals 0 because the second term sums to 0.

4 Python Code For Generating Monomials

```
import csv
from itertools import chain, combinations
#new: calc betas, get_i_c, get_c_from_subword_index, get_root
#-----------------HELPER FUNCTIONS-----------------
def rev(x):
    return x[::-1] #simple helper function to reverse list
def write_element_correctly(ele):
    while ele[0]!=min(ele):
        ele = ele[1:]+[ele[0]]
    return ele
def powerset(iterable):
    "powerset([1,2,3]) --> () (1,) (2,) (3,) (1,2) (1,3) (2,3) (1,2,3)"
    s = list(iterable)
    return chain.from_iterable([list(x) for x in combinations(s, r)] for r in
#--------------------------------------------------
def create_csv_for_long_word(long_word):
    filename = ",".join([str(x) for x in long_word])
    with open("/Users/lucasfagan/Desktop/PRUV code/"+filename.replace(",","")-
        writer = csv.writer(csvfile)
        writer.writerow(['Long word','i','v^-1','subword','indices','c'])
        v_inv_list = [find_v_inverse(long_word, i) for i in range(1,max(long_w
        for i in range(len(v_inv_list)):
            indx, subwords = get_subwords(long_word,v_inv_list[i])
            for j in range(len(indx)):
                c = get_c_from_subword_index(subwords[j],indx[j],long_word,i+
                writer.writerow([long_word,i+1,v_inv_list[i],subwords[j],indx
    print("Created csv for "+filename)
def get_i_c(long_word):
    final_list = []
    v_inv_list = [find_v_inverse(long_word, i) for i in range(1,max(long_word
    for i in range(len(v_inv_list)):
        indx, subwords = get_subwords(long_word,v_inv_list[i])
        for j in range(len(indx)):
            c = get_c_from_subword_index(subwords[j],indx[j],long_word,i+1)
            final_list.append((i+1,c))
    return final_list
def get_c_from_subword_index(subword, indx, long_word,i):
    c=[]
    for x in range(len(long_word)):
        if x in indx:
            if len(c)==0 and subword[indx.index(x)]!=i:
                c.append(0)
            else:
```

```
                c.append(1)
        else:
            c.append(0)
    return c
def create_csv_for_gl(n):
    with open("/Users/lucasfagan/Desktop/PRUV code/GL_"+str(n)+".csv", "w", n
        writer = csv.writer(csvfile)
        writer.writerow(['Long word','i','v^-1','# subwords','subwords'])
        writer.writerow([])
        if n<=5:
            long_words = generate_long_words(n,10000)
        else:
            long_words = generate_long_words(n,100)
        first_lw = long_words[0] #O(1) access
        v_inv_list = [find_v_inverse(first_lw, i) for i in range(1,n)]
        for word in long_words:
            for i in range(1,n):
                v_inv = v_inv_list[i-1]
                indx, subw = get_subwords(word, v_inv)
                writer.writerow([word, i, v_inv, len(subw), subw])
            writer.writerow([])
    print("Generated csv for GL_{}".format(n))
def generate_long_words(n, num_words):
    #creates long word by [1,2,1,3,2,1...n,n-1,...,2,1]
    first=[]
    for k in range(1,n):
        a = list(range(1,k+1))
        a.reverse()
        first.extend(a)
    #print("first",first)
    all_long_words = recurse([first],first, num_words)
    return all_long_words
    #print(all_long_words)
    #print(len(all_long_words))
    #print_all_v_inverse_for_choice_of_lw(first)
    # for long_word in all_long_words:
    #     print(get_element(long_word))
def recurse(words, lw, num_words):
    if len(words)>=num_words:
        return words
    #lw=long word
    for i in range(len(lw)-2):
        #print(i)
        if lw[i]==lw[i+2] and (lw[i+1]-1==lw[i] or lw[i+1]+1==lw[i]): #can be
            #print("can be braided")
            lw_braided = lw[:i]+[lw[i+1],lw[i],lw[i+1]]+lw[i+3:]
```

```
                #print(lw_braided)
                #print(words)
                if lw_braided not in words:
                    #print("can be braided", lw_braided)
                    words=recurse(words+[lw_braided], lw_braided, num_words) #bra
        for i in range(len(lw)-1):
            if lw[i]-lw[i+1]>1 or lw[i]-lw[i+1]<-1: #|i-j|>1
                lw_switched = lw[:i]+[lw[i+1],lw[i]]+lw[i+2:]
                if lw_switched not in words:
                    #print("can be switched", lw_switched)
                    words=recurse(words+[lw_switched], lw_switched, num_words)
        return words
        #get_element(first)
    def find_v_inverse(long_word,i):
        #find v inverse such that w_0*s_i*lambda_i=v_i*lambda_i
        long_word = long_word + [i] #add s_i to w_0
        long_word = rev(long_word) #reverse it in order to work L->R
        reduced_forms = recurse_vinv([long_word],long_word, i)
        #print(reduced_forms)
        min_len = min([len(x) for x in reduced_forms])
        for rf in reduced_forms:
            if len(rf)==min_len:
                return rf
    def recurse_vinv(reduced_forms, cur, i):
        if len(cur)<=1:
            reduced_forms.append(cur)
            return reduced_forms
        if cur[0]!=i: #first remove all the s_j such that i =/= j with lambda_i
            cur=cur[1:]
            reduced_forms.append(cur)
            reduced_forms = recurse_vinv(reduced_forms, cur, i)
        for j in range(len(cur)-1): #then remove all of the repeated elements (s s
            if cur[j]==cur[j+1]:
                temp = cur[:j] + cur[j+2:]
                reduced_forms.append(temp)
                reduced_forms = recurse_vinv(reduced_forms, temp, i)
        for j in range(len(cur)-1): #then try all of the |i-j|>1
            if cur[j]-cur[j+1]>1 or cur[j]-cur[j+1]<-1:
                temp = cur[:j]+[cur[j+1],cur[j]]+cur[j+2:]
                if temp not in reduced_forms:
                    reduced_forms.append(temp)
                    reduced_forms = recurse_vinv(reduced_forms, temp, i)
        for j in range(len(cur)-2): #then try all of the braid relations
            if cur[j]==cur[j+2] and (cur[j+1]-1==cur[j] or cur[j+1]+1==cur[j]):
                temp = cur[:j]+[cur[j+1],cur[j],cur[j+1]]+cur[j+3:]
                if temp not in reduced_forms:
```

```
                reduced_forms.append(temp)
                reduced_forms = recurse_vinv(reduced_forms, temp, i)
    return reduced_forms
def print_all_v_inverse_for_choice_of_lw(long_word):
    for j in range(1,max(long_word)+1):
        print("v^-1 =",find_v_inverse(long_word,j),"for lambda_"+str(j))
def get_subwords(long_word, vi_inv):
    # all_subwords = []
    # for i in range(len(long_word)):
    #     for j in range(i+1,len(long_word)+1):
    #         all_subwords.append(long_word[i:j])
    # #print(all_subwords)
    all_indices = list(powerset(list(range(len(long_word)))))[1:]
    #all_subwords = list(powerset(long_word))[1:]
    subwords_indices = [x for x in all_indices if get_element([long_word[y] fo
    minlen = min([len(x) for x in subwords_indices])
    reduced_subwords_indices = [x for x in subwords_indices if len(x)==minlen
    return reduced_subwords_indices,[[long_word[y] for y in x] for x in reduc
    # subwords_element_dict = {tuple(subword):get_element(subword) for subword
    # print(subwords_element_dict)
    # element_to_subwords_dict = {tuple(ele):[x for x in subwords_element_dic
    # return create_subwords_dict
def get_element(word):
    #parameters: word (list), e.g. [1,3,2] refers to s_1 s_3 s_2 = (1 2)(3 4)
    starter = list(range(1,max(word)+2))
    word=rev(word)
    resultant_places = []
    for num in starter:
        cur = num
        for refl in word:
            if cur==refl:
                cur+=1
            elif cur==refl+1:
                cur-=1
        resultant_places.append(cur)
        #print(num," goes to",cur)

    final_element = []

    for num in starter:
        ele = [num]
        x = resultant_places[num-1] #-1 to adjust for array starting at 0
        while x!=num:
            ele.append(x)
            x=resultant_places[x-1]
        correct_ele = write_element_correctly(ele)
```

```
        if correct_ele not in final_element and len(correct_ele)>1:
            final_element.append(correct_ele)

    return final_element
def get_root(reflections, a):
    maxnum = max(reflections+[a])
    alpha_final = [0]*maxnum
    alpha_final[a-1]=1
    #right now we have just alpha_i

    reflections = rev(reflections)
    for ref in reflections: #iterating backwards
        #print(alpha_final)
        tot = 0
        if ref-2>=0:
            tot+=alpha_final[ref-2]
        if ref<len(alpha_final):
            tot+=alpha_final[ref]
        tot-=alpha_final[ref-1]
        alpha_final[ref-1]=tot


    return alpha_final
def print_betas(long_word):
    print("a = alpha, b = beta")
    betas = [get_root(long_word[:i],long_word[i]) for i in range(len(long_word
    beta_strings = []
    for beta in betas:
        temp = ""
        for i,num_alpha in enumerate(beta):
            temp+=("a_"+str(i+1)+" + ")*num_alpha
        beta_strings.append(temp[:-3])
    for i, beta_str in enumerate(beta_strings):
        print("b_"+str(i+1)+" = "+beta_str+" dual")
def calc_denominators(long_word):
    to_return = []
    betas = [get_root(long_word[:i],long_word[i]) for i in range(len(long_word
    for i in range(max(long_word)):
        to_return.append([1 if len(beta)>=i+1 and beta[i]>0 else 0 for beta in
    return to_return
def get_bs(long_word):
    numerators = get_i_c(long_word)
    denoms = calc_denominators(long_word)
    to_return = []
    for i in range(1,max(long_word)+1):
        relevant_numerators = [x[1] for x in numerators if x[0]==i]
```

```
        terms = [[num - denom for num, denom in zip(rel_num,denoms[i-1])] for
        to_return.append(terms)
    return to_return
def print_bs(long_word):
    bs = get_bs(long_word)
    for j,b in enumerate(bs):
        s = ""
        for term in b:
            s+=" ".join(["b_"+str(i+1) for i,x in enumerate(term) if x==1]) i
            s+=" / "
            s+=" ".join(["b_"+str(i+1) for i,x in enumerate(term) if x==-1])
            s+=" + "
        print("s_"+str(j+1)+" = "+s[:-3])
```