# BRAID VARIETIES (III)

2026/01/17

From now on, we will assume $G$ to be simply-laced, i.e. ADE type. We will denote

$$X(\beta) = X(\beta, w_0), \qquad w_0 = \text{Demazure product of } \beta.$$

By an example discussed in the first note, we can assume $w_0$ to be the longest element of $W$.

## 1. Demazure waves

1.1. **Demazure waves.** For a braid word $\beta = \sigma_{i_1} \dots \sigma_{i_\ell}$, recall the dense stratification described last time is parametrized by the sequence

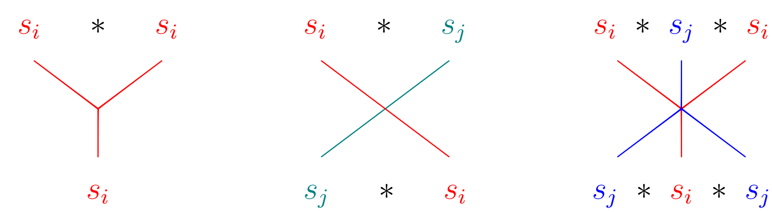
$$(\cdots, \quad s_{i_{\ell-2}} s_{i_{\ell-1}} s_{i_\ell}, \quad s_{i_{\ell-1}} s_{i_\ell}, \quad s_{i_\ell}, \quad \mathsf{id}).$$

This could be viewed the algorithm of computing the Demazure product $s_{i_1} * \cdots * s_{i_\ell}$ (from right to left). Besides of this way, thanks to the associativity of Demazure product, there are different ways. For example,

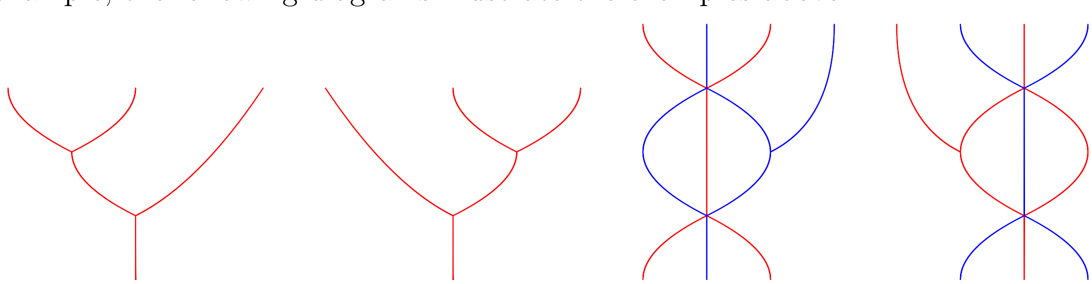
$$\begin{array}{cc}
(\text{type } A_1) & (\text{type } A_2) \\
(\sigma_1\sigma_1)\sigma_1 = \sigma_1\sigma_1 = \sigma_1 & (\sigma_1\sigma_2\sigma_1)\sigma_2 = \sigma_2\sigma_1(\sigma_2\sigma_2) = \sigma_2\sigma_1\sigma_2 = \sigma_1\sigma_2\sigma_1 \\
\sigma_1(\sigma_1\sigma_1) = \sigma_1\sigma_1 = \sigma_1, & \sigma_1(\sigma_2\sigma_1\sigma_2) = (\sigma_1\sigma_1)\sigma_2\sigma_1 = \sigma_1\sigma_2\sigma_1 = \sigma_2\sigma_1\sigma_2.
\end{array}$$

Demazure wave is a way of remember different ways.

A **Demazure wave** consists of




For example, the following diagrams illustrate the examples above.



1.2. **Wave varieties.** For a wave $\mathfrak{W} : \beta_1 \to \beta_2$, we can define **wave variety**

$$X(\mathfrak{W}) = \left\{ (g_r B)_{r \in \mathsf{Region}(\mathfrak{W})} \;\middle|\; \begin{array}{c} a \overset{i}{\mid} b \Rightarrow g_b B \xrightarrow{s_i} g_a B \\ gB_{\text{leftmost}} = B, gB_{\text{rightmost}} = w_0 B \end{array} \right\}.$$

For example, when $\mathfrak{W} : \beta \to \beta$ consists of only vertical edges, $X(\mathfrak{W})$ is nothing but the braid variety $X(\beta)$. It is not hard to obtain the following local behavior at each vertex of a wave:

$$\left\{ \begin{tikzcd} & gB \arrow[dr, "s_i"] & \\ g_1 B \arrow[ur, "s_i"] \arrow[rr, "s_i"'] & & g_2 B \end{tikzcd} \right\} \begin{array}{l} \xrightarrow{\mathbb{C}^\times\text{-bundle}} \{g_1 B \xrightarrow{s_i} g_2 B\} \\ \xrightarrow{\text{open}} \{g_1 B \xrightarrow{s_i} gB \xrightarrow{s_i} g_2 B\} \end{array}$$

When $m_{ij} = 2$,

$$\left\{ \begin{tikzcd} & gB \arrow[dr, "s_j"] & \\ g_1 B \arrow[ur, "s_i"] \arrow[dr, "s_j"'] & & g_2 B \\ & g'B \arrow[ur, "s_i"'] & \end{tikzcd} \right\} \begin{array}{l} \xrightarrow{\sim} \{g_1 B \to gB \to g_2 B\} \\ \xrightarrow{\sim} \{g_1 B \to g'B \to g_2 B\} \end{array}$$

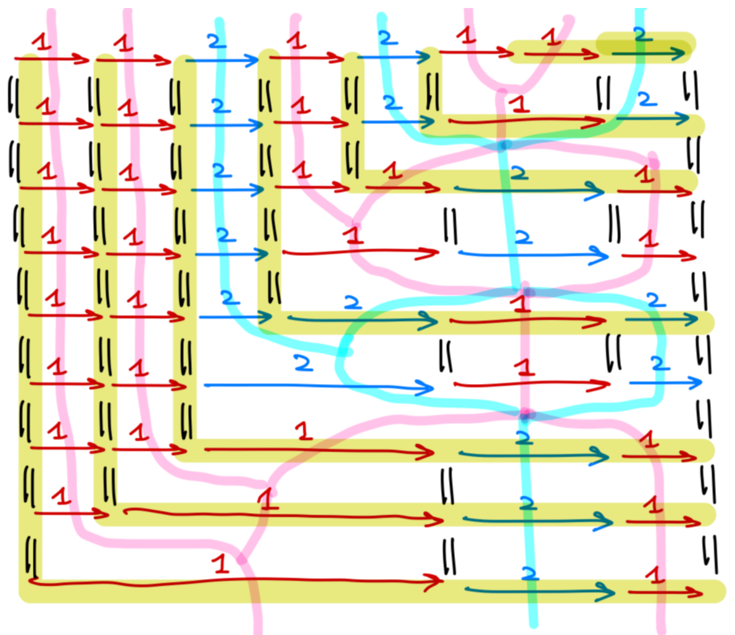
When $m_{ij} = 3$

$$\left\{ \begin{tikzcd} & gB \arrow[r, "s_j"] & g'B \arrow[dr, "s_i"] & \\ g_1 B \arrow[ur, "s_i"] \arrow[dr, "s_j"'] & & & g_2 B \\ & g''B \arrow[r, "s_i"'] & g'''B \arrow[ur, "s_j"'] & \end{tikzcd} \right\} \begin{array}{l} \xrightarrow{\sim} \{g_1 B \to gB \to g'B \to g_2 B\} \\ \xrightarrow{\sim} \{g_1 B \to g''B \to g'''B \to g_2 B\} \end{array}$$

**Theorem.** Let $\mathfrak{W} : \beta_1 \to \beta_2$ be a Demazure wave.

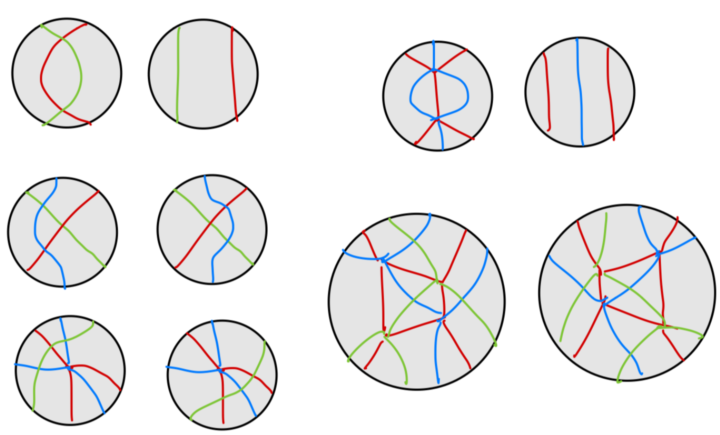
- the forgetful map
$$X(\mathfrak{W}) \to X(\beta_1)$$
is always an open embedding;
- if $\mathfrak{W}$ consists only of braid moves, the upper and lower projections are both isomorphisms:
$$X(\mathfrak{W}) \xrightarrow{\sim} X(\beta_i) \qquad i = 1, 2.$$
In particular, they give an isomorphism $X(\beta_1) \cong X(\beta_2)$.

**Example.** The wave corresponds to the algorithm of computing the Demazure product from right to left is called **inductive wave**. The maximal stratum $X(\beta, \underline{\mathbf{u}})$ is the image of the inductive $\mathfrak{W}$.

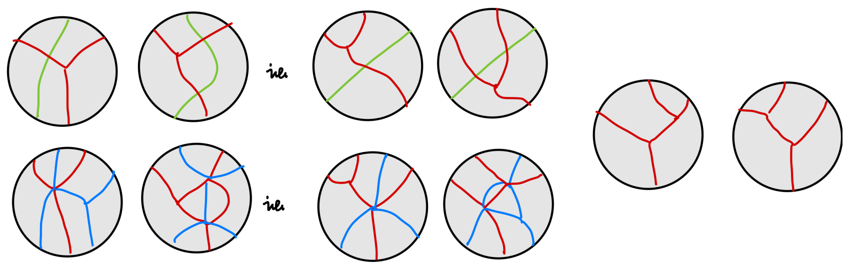


**Lemma.** Let $\beta_1, \beta_2$ be two braid words.

(1) Different waves consisting only of braid moves are related by **two-braid (Zamolodchikov) relations**



(2) In general, different waves are related by above relations as well as the **1212 relations** (left) and **mutations** (right)



*Proof.* Consider all possible positions in a braid word where one can apply $\sigma_i\sigma_i \to \sigma_i$ the braid moves. If such positions do not overlap, the operations commute. If they overlap, then these involves at most 3 different simple reflections. So it reduces to $A_1^2, A_2 \times A_1, A_3$. In type $A$, this is a theorem of Elias. $\square$

**Theorem.** Let $\beta_1, \beta_2$ be two braid words. For two waves $\mathfrak{W}_1$ and $\mathfrak{W}_2$ related by two-braid relations and 1212 relations, $X(\mathfrak{W}_1)$ and $X(\mathfrak{W}_2)$ give the same open subset in $X(\beta_1)$.

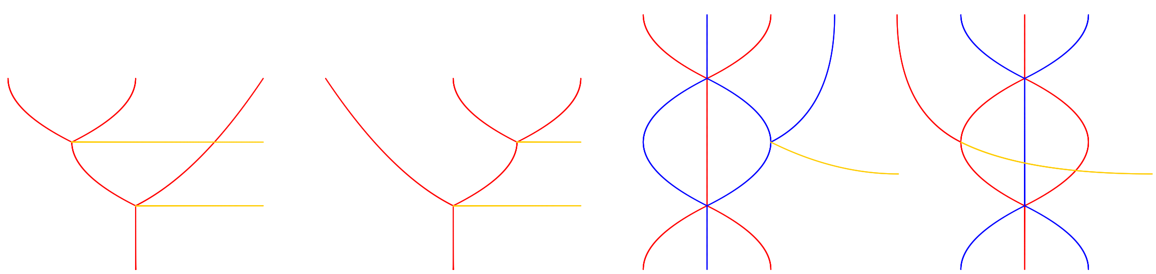
To prove this theorem, we need to parametrize $X(\mathfrak{W})$, as we did for braid variety $X(\beta)$.

## 2. Parametrization

Recall that

$$B_i(z) = x_i(z)\dot{s}_i = \text{image of } \begin{bmatrix} z & -1 \\ 1 & 0 \end{bmatrix} \in G.$$

Let $\mathfrak{W}^=$ be the diagram obtained by adding dashed edges to trivalent vertices.



Note that there are more regions in $\mathfrak{W}^=$ than $\mathfrak{W}$. We define the **parametrization**

$$X(\mathfrak{W}^=) = \left\{ (g_r)_{r \in \mathsf{Region}(\mathfrak{W}^=)} \;\middle|\; \begin{array}{c} a \overset{i}{\mid} b \Rightarrow g_b = g_a B_i(z_e) \text{ for some } z_e \\ \overset{a}{\underset{b}{\cdots}} \Rightarrow g_a = g_b U_e \text{ for some } U_e \in B \\ g_{\text{leftmost}} = 1, g_{\text{rightmost}} \in w_0 B \end{array} \right\}.$$

In particular, we can also identify

$$X(\mathfrak{W}^=) = \left\{ (z_e)_{e \in \mathsf{Edge}(\mathfrak{W}^=)} \times (U_d)_{d \in \mathsf{Dash}(\mathfrak{W}^=)} \;\middle|\; \begin{array}{c} z_e \in \mathbb{C}, U_d \in B \\ (\textbf{Comm}) \text{ and } (\textbf{Right}) \end{array} \right\}.$$

Here (**Comm**) means at each vertex, a corresponding diagram commutes; see Appendix. Since the diagram commutes, it is well-defined to associate $g_r$ for each region $r$ of $\mathfrak{W}^{=}$, (**Right**) means the rightmost region is in $w_0B$.

**Proposition.** The natural projection

$$X(\mathfrak{W}^{=}) \to X(\mathfrak{W}), \qquad (g_r) \mapsto (g_rB)$$

is an isomorphism.

*Proof.* This map is obviously surjective. Let us prove it is injective. At each vertex, the top/left labeling determines the bottom/right labeling (computed in Appendix). Thus the top $z$-labeling determines all $z$-labeling. This corresponds to a unique element in $X(\beta)$ for $\beta$ the top braid. $\square$

**Example.** Consider the two waves above.

$$\begin{tikzcd}
g_0 \arrow[r, red, "B_1(z_1)"] \arrow[d, equal] & g_1 \arrow[r, blue, "B_2(z_2)"] & g_2 \arrow[r, red, "B_1(z_3)"] & g_3 \arrow[r, blue, "B_2(z_4)"] \arrow[d, equal] & g_4 \arrow[d, equal] \\
g_0 \arrow[r, blue, "B_2(z_3)"] \arrow[d, equal] & g_4 \arrow[r, red, "B_1(z_1z_3-z_2)"] \arrow[d, equal] & g_5 \arrow[r, blue, "B_2(z_1)"] \arrow[d, equal] & g_3 \arrow[r, blue, "B_2(z_4)"] & g_4 \\
g_0 \arrow[r, blue, "B_2(z_3)"] \arrow[d, equal] & g_4 \arrow[r, red, "B_1(z_1z_3-z_2)"] & g_5 \arrow[rr, blue, "B_2(z_1-1/z_4)"] & & g_4 \arrow[u, "\begin{bmatrix} 1 & 0 & 0 \\ 0 & z_4 & -1 \\ 0 & 0 & z_4^{-1} \end{bmatrix}"'] \arrow[d, equal] \\
g_0 \arrow[r, red, "B_1(z_1-1/z_4)"] & g_6 \arrow[r, blue, "B_2(z_2-z_3/z_4)"] & g_7 \arrow[rr, red, "B_1(z_3)"] & & g_4
\end{tikzcd}$$

$$\begin{tikzcd}
g_0 \arrow[r, red, "B_1(z_1)"] \arrow[d, equal] & g_1 \arrow[r, blue, "B_2(z_2)"] \arrow[d, equal] & g_2 \arrow[r, red, "B_1(z_3)"] & g_3 \arrow[r, blue, "B_2(z_4)"] & g_4 \arrow[d, equal] \\
g_0 \arrow[r, red, "B_1(z_1)"] \arrow[d, equal] & g_1 \arrow[r, red, "B_1(z_4)"] & g_4 \arrow[r, blue, "B_2(z_2z_4-z_3)"] & g_5 \arrow[r, red, "B_1(z_2)"] & g_4 \\
g_0 \arrow[rr, red, "B_1(z_1-1/z_4)"] \arrow[d, equal] & & g_4 \arrow[u, "\begin{bmatrix} z_4 & -1 & 0 \\ 0 & z_4^{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix}"'] \arrow[r, blue, "B_2(z_2-z_3/z_4)"] & g_5 \arrow[u, "\begin{bmatrix} z_4 & -z_2z_4+z_3 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & z_4^{-1} \end{bmatrix}"] \arrow[r, red, "B_1(z_3)"] & g_4 \arrow[u, "\begin{bmatrix} 1 & 0 & 0 \\ 0 & z_4 & -1 \\ 0 & 0 & z_4^{-1} \end{bmatrix}"] \arrow[d, equal] \\
g_0 \arrow[rr, blue, "B_2(z_3)"] & & g_6 \arrow[r, red, "B_1(z_1z_3-z_2)"] & g_7 \arrow[r, blue, "B_2(z_1-1/z_3)"] & g_4
\end{tikzcd}$$

**Proposition.** Fix a reduced word $w_0$.

(1) For a given top $z$-labeling, the bottom $z$-labeling do not depend on the choice of Demazure wave $\beta \to \sigma_{w_0}$.

(2) The condition (**End**) is equivalent to the vanishing of the bottom $z$-labeling.

(3) The condition (**Comm**) is equivalent to the nonvanishing of northeast $z$-labeling of trivalent vertices.

*Proof.* For any reduced word $w_0 = s_{i_1} \cdots s_{i_l}$, we have an isomorphism

$$\mathbb{C}^{\ell(w_0)} \xrightarrow{\sim} Bw_0B/B, \qquad (z'_k) \mapsto B_{i_1}(z'_1) \cdots B_{i_l}(z'_l)B.$$

(1) As a result, the bottom $z$-labeling is given by the unique $(z'_k)$ mapped to the flag parametrized by the product of top $B$-matrices. (2) In particular, only the vanishing $(z'_k)$ could satisfy (**End**). (3) is clear. $\square$

**Example.** We remark that mutation does not give the same open subsets. Consider $\sigma_1^3 \to \sigma_1$.

$$
\begin{tikzcd}
g_0 \arrow[r, "B(z_1)"] \arrow[d, equal] & g_1 \arrow[r, "B(z_2)"] & g_2 \arrow[r, "B(z_3)"] & g_3 \\
g_0 \arrow[rr, "B(z_1-z_2^{-1})"] \arrow[d, equal] & & g_2 \arrow[r, "B(z_2^2z_3-z_2)"] \arrow[u, "\begin{bmatrix} z_2 & -1 \\ 0 & z_2^{-1} \end{bmatrix}"'] & g_3 \arrow[u, "\begin{bmatrix} z_2^{-1} & 0 \\ 0 & z_2 \end{bmatrix}"'] \\
g_0 \arrow[rrr, "B\left(\frac{z_1z_2z_3-z_1-z_3}{z_2z_3-1}\right)"] & & & g_3 \arrow[u, "\begin{bmatrix} z_2^2z_3-z_2 & -1 \\ 0 & (z_2^2z_3-z_2)^{-1} \end{bmatrix}"']
\end{tikzcd}
\qquad
\begin{tikzcd}
g_0 \arrow[r, "B(z_1)"] \arrow[d, equal] & g_1 \arrow[r, "B(z_2)"] \arrow[d, equal] & g_2 \arrow[r, "B(z_3)"] & g_3 \\
g_0 \arrow[r, "B(z_1)"] \arrow[d, equal] & g_1 \arrow[rr, "B(z_2-z_3^{-1})"] & & g_3 \arrow[u, "\begin{bmatrix} z_3 & -1 \\ 0 & z_3^{-1} \end{bmatrix}"'] \\
g_0 \arrow[rrr, "B\left(\frac{z_1z_2z_3-z_1-z_3}{z_2z_3-1}\right)"] & & & g_3 \arrow[u, "\begin{bmatrix} z_2-z_3^{-1} & -1 \\ 0 & (z_2-z_3^{-1})^{-1} \end{bmatrix}"']
\end{tikzcd}
$$

The open subsets are described as

$$z_2 \neq 0, z_2^3z_3 - z_2 \neq 0, \qquad \text{v.s.} \qquad z_3 \neq 0, z_2 - z_3^{-1} \neq 0.$$

*Proof of the theorem.* It suffices to check two waves related by a single Zamolodchikov relation or 1212 relation. By picking dashed edges $\mathfrak{W}^=$ avoiding their difference, we will define an isomorphism over $X(\mathfrak{W}_1^=) \to X(\mathfrak{W}_2^=)$ such that, if we write $(z_{1e}) \mapsto (z_{2e})$, then $z_{2e} = z_{1e}$ for edges $e$ not inside the difference (i.e. edges outside of the difference and edges $e$ intersecting the boundary of the difference). Such an isomorphism will be the required isomorphism.

By the solution of (**Comm**), we can express $z_e$ for $e \in \mathsf{inside} \cup \mathsf{lower}$ from $z_e$ for $e \in \mathsf{upper}$. This will determine an isomorphism, and it remains to check it is well-defined. That is,

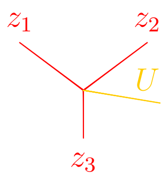
(1) $z_e$ for edges $e$ intersecting the lower boundary agrees;
(2) no new pole of $z_e$ was introduced.

For two-braid relations, since no pole is introduced, so it suffices to check (1). This can be done by computation, but there is a quicker way of seeing this. Noting that the lower boundary of each two-braid is a reduced word in Weyl group, by (1) of Proposition above, (1) is true.

For 1212 relations, (1) is similar. But in this case, a pole is introduced. From the example above, the pole is the same, given by $z_4 \neq 0$. This proves (2). $\square$

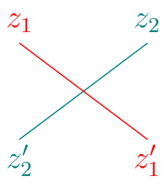
## Appendix A. Explicit Solutions

We can solve the equations explicitly.



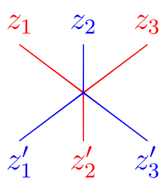

$$\begin{tikzcd} {} \arrow[r, "B_i(z_1)"] \arrow[d, equal] & {} \arrow[r, "B_i(z_2)"] & {} \\ {} \arrow[rr, "B_i(z_3)"] & & {} \arrow[u, "U"'] \end{tikzcd}$$

$$z_3 = z_1 - z_2^{-1}$$
$$U = \text{image of } \begin{bmatrix} z_2 & -1 \\ 0 & z_2^{-1} \end{bmatrix}$$



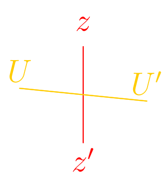

$$\begin{tikzcd} {} \arrow[r, "B_i(z_1)"] \arrow[d, equal] & {} \arrow[r, "B_j(z_2)"] & {} \arrow[d, equal] \\ {} \arrow[r, "B_j(z_2')"] & {} \arrow[r, "B_i(z_1')"] & {} \end{tikzcd}$$

$$z_1' = z_1$$
$$z_2' = z_2$$




$$\begin{tikzcd} {} \arrow[r, "B_i(z_1)"] \arrow[d, equal] & {} \arrow[r, "B_j(z_2)"] & {} \arrow[r, "B_i(z_3)"] & {} \arrow[d, equal] \\ {} \arrow[r, "B_j(z_1')"] & {} \arrow[r, "B_i(z_2')"] & {} \arrow[r, "B_j(z_3')"] & {} \end{tikzcd}$$

$$z_1' = z_3$$
$$z_3' = z_1$$
$$z_2' = z_1 z_3 - z_2$$




$$\begin{tikzcd} {} \arrow[r, "B_i(z)"] & {} \\ {} \arrow[u, "U"] \arrow[r, "B_i(z')"] & {} \arrow[u, "U'"'] \end{tikzcd}$$

$$z' = z \cdot \alpha_i(U) + \xi_i(U)$$
$$U' = B_i(z')^{-1} U B_i(z) \in B$$

Let us explain $\alpha_i$ and $\xi_i$. We can write $U = U_1 U_2$ for $U_1 \in \mathrm{Rad}(B)$ and $U_2 \in T$. Then

$$\xi_i(U) = \text{coefficient of } E_i \text{ in } U_1, \qquad \eta_i(U) = \alpha_i(U).$$

For example, in $GL_2$,

$$\begin{bmatrix} a & b \\ & d \end{bmatrix} = \begin{bmatrix} 1 & b/d \\ & 1 \end{bmatrix} \begin{bmatrix} a & \\ & d \end{bmatrix} \begin{array}{l} \xmapsto{\xi} b/d, \\ \xmapsto{\eta} a/d. \end{array}$$

All the relations follow from the computation in $SL_2 \subset GL_2$, $SL_2 \times SL_2 \subset GL_4$, $SL_3 \subset GL_3$ and $SL_2 \subset GL_2$. The following is the code.

```
R.<z1,z2,z3,a,b,c> = QQ[];
B = lambda z: matrix([[z,-1],[1,0]])
U = matrix([[a,b],[0,c]]);
Rel = (B(z1)*B(z2) - B(z3)*U).change_ring(SR)
solve([Rel[i][j]==0 for i in range(2) for j in range(2)], SR.var("z3,a,b,c"))
```

```
[[z3 == (z1*z2 - 1)/z2, a == z2, b == -1, c == (1/z2)]]
```

```
R.<z1,z2,z1p,z2p> = QQ[]
B1 = lambda z: matrix([[z,-1,0,0],[1,0,0,0],[0,0,1,0],[0,0,0,1]])
B3 = lambda z: matrix([[1,0,0,0],[0,1,0,0],[0,0,z,-1],[0,0,1,0]])
Rel = (B1(z1)*B3(z2)-B3(z2p)*B1(z1p)).change_ring(SR)
solve([Rel[i][j]==0 for i in range(4) for j in range(4)], SR.var("z1p,z2p"))
```

```
[[z1p == z1, z2p == r1]]
```

```
R.<z1,z2,z3,z1p,z2p,z3p> = QQ[]
B1 = lambda z: matrix([[z,-1,0],[1,0,0],[0,0,1]])
B2 = lambda z: matrix([[1,0,0],[0,z,-1],[0,1,0]])
Rel = (B1(z1)*B2(z2)*B1(z3)-B2(z1p)*B1(z2p)*B2(z3p)).change_ring(SR)
solve([Rel[i][j]==0 for i in range(2) for j in range(2)], SR.var("z1p,z2p,z3p"))
```

```
[[z1p == z3, z2p == z1*z3 - z2, z3p == z1]]
```

```
R.<z,zp,a,b,c,ap,bp,cp> = QQ[]
B = lambda z: matrix([[z,-1],[1,0]])
U = matrix([[a,b],[0,c]]); Up = matrix([[ap,bp],[0,cp]])
Rel = (U*B(z)-B(zp)*Up).change_ring(SR)
solve([Rel[i][j]==0 for i in range(2) for j in range(2)], SR.var("zp,ap,bp,cp"))
```

```
[[zp == (a*z + b)/c, ap == c, bp == 0, cp == a]]
```