REDUCTIONS OF YOUNG TABLEAU BIJECTIONS

YIWEI FU

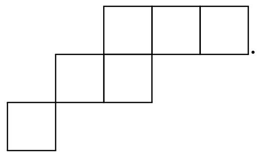
1. Introduction

Computational complexity of the bijections between combinatorial objects is an intriguing yet very sparsely investigated subject. [3] introduces the linear equivalences between several combinatorial bijections, thus unifying them under the computational point of view. This provides a way to classify these bijections and shed lights on how to systematically study these bijections in the future.

2. Background and Notation

2.1. Diagrams and Tableaux.

Definition 2.1. Suppose two partitions λ, μ with $\lambda_i > \mu_i$ for all i. A standard Young diagram is denoted $[\lambda]$; a skewed Young diagram of shape λ/μ is denoted $[\lambda/\mu]$.

Example 2.1. Let $\lambda = (5, 4, 2), \mu = (2, 1)$, then $[\lambda/\mu]$ is .

Definition 2.2. Given skewed or standard Young diagram of shape $[\lambda/\mu]$ or $[\lambda]$, a tableau is a filling of the boxes in positive integers such that it is weakly increasing in rows and strictly increasing in columns. The set $\mathrm{YT}(\lambda/\mu, \mathbf{a})$ or $\mathrm{YT}(\lambda, \mathbf{a})$ contains all the tableaux of such shape so that for all $A \in \mathrm{YT}(\lambda/\mu, \mathbf{a})$ or $\mathrm{YT}(\lambda, \mathbf{a})$, $\mathbf{a} = (a_1, \dots, a_n) =:$ $\mathbf{weight}(A)$ has a_j counting the number of j's in the tableau A.

Definition 2.3. A sequence $\mathbf{i} = (i_1, \dots, i_n)$ is *positive* if the number of j's in $\mathbf{i}$ is weakly decreasing. It is *dominant* if all the subsequences containing first k elements for $1 \le k \le n$ are positive.

Example 2.2. $(1, 1, 2, 3, 4)$ is positive and dominant. $(1, 2, 2, 1, 1)$ is positive but not dominant since subsequence $(1, 2, 2)$ is not positive.

Definition 2.4. A *Littlewood-Richardson* (LR) tableau is a tableau A such that the sequence (called $\mathbf{word}(A)$) obtained by reading the tableau **right-to-left** each row and then top-to-bottom is *dominant*. Denote all the LR-tableaux of shape λ/μ and weight ν as $\mathrm{LR}(\lambda/\mu, \nu)$.

Example 2.3.

$$\mathbf{word}\left(\begin{array}{cccc} & \boxed{1} & \boxed{2} & \boxed{5} \\ \boxed{2} & \boxed{3} & \boxed{4} & \end{array}\right) = (5, 2, 1, 4, 3, 2).$$

Only LR-tableau of standard Young diagram λ is to fill every i-th row with i, which is called *canonical* tableau, denoted $\mathrm{Can}(\lambda)$.

$$\mathrm{Can}((5,3,2)) = \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 1 \\ \hline 2 & 2 & 2 \\ \cline{1-3} 3 & 3 \\ \cline{1-2}\end{array}$$

2.2. **Maps and Bijections.**

Definition 2.5 (RSK)**.** The *RSK* map φ, as introduced in class, establishes a bijection between

$$\left\{\mathbb{Z}_{\geq 0}\text{-matrices} \;\middle|\; \begin{array}{c}\text{row sum} = \mathbf{a} \\ \text{column sum} = \mathbf{b}\end{array}\right\} \leftrightarrow \left\{(T,U) \;\middle|\; \begin{array}{c}T \in \mathrm{YT}(\lambda, \mathbf{a}) \\ U \in \mathrm{YT}(\lambda, \mathbf{b})\end{array} \text{ for some } \lambda\right\}.$$

Here we fix matrix size and the max entry of SSYT to be some k.

Definition 2.6 (Jeu de Taquin)**.** *Jeu de Taquin* ψ is a map from skewed semistandard Young tableaux to semistandard Young tableaux by the following methods:

Given a skewed tableau, choose a top-left border piece and move the blocks one at a time so that after a series of moves we also get a skewed tableau.

This map is neither into nor onto.

Example 2.4. Figure 1 shows one step in the Jeu de Taquin map. Starting from a tableau, there are different sequences of steps, but we can use the diamond lemma to show that game defined by Jeu de Taquin is confluent (i.e. Jeu de Taquin map is well-defined).

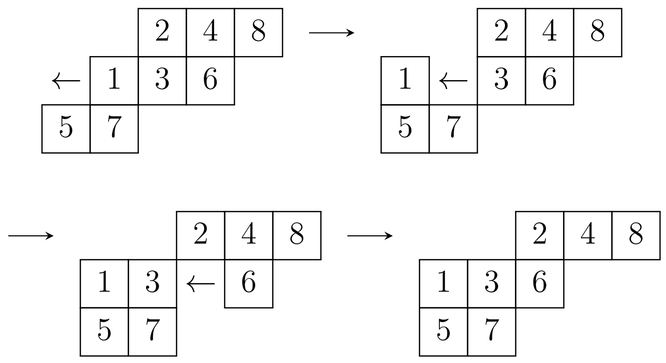


FIGURE 1. One step in a Jeu de Taquin game

Definition 2.7. The *Littlewood-Robinson map* ϕ is a bijection between skewed semistandard Young tableaux and pairs of SSYT and LR-tableaux.

Definition 2.8. The *Bender-Knuth (BK) transformations* s_r is defined by follows:

- Given $A \in \mathrm{YT}(\lambda/\mu, \mathbf{a})$, let $(a_{i,j})$ be the Gelfand-Tsetlin(GT)-pattern.

- Let $B = s_r(A)$ be a Young tableau such that the corresponding GT-pattern $(b_{i,j})$ is defined as follows:

$$b_{i,j} = \begin{cases} \min\{a_{i,r+1}, a_{i-1,r-1}\} + \max\{a_{i,r-1}, a_{i+1,r+1} - a_{i,r}\} - a_{i,r} & j = r \\ a_{i,j} & \text{otherwise} \end{cases}$$

Remark. We notice that

$$s_r : \mathrm{YT}(\lambda/\mu, \mathbf{a}) \to \mathrm{YT}(\lambda/\mu, \mathbf{a}')$$

where $\mathbf{a}'$ is obtained by switching the r-th and $r+1$-th element. We have $(s_r)^2 = 1$ and $s_i s_j = s_j s_i$ for $|i-j| \geq 2$.

Definition 2.9. Define two transformations $t_{r,m-r}, z_m$ for $1 \leq r < m$ as follows:

$$\begin{aligned} z_m &= (s_1)(s_2 s_1)\dots(s_{m-1}\dots s_2 s_1) \\ t_{r,m-r} &= (s_{m-r}s_{m-r+1}\dots s_{m-1})\dots(s_2 s_3 \dots s_{r+1})(s_1 s_2 \dots s_r) \end{aligned}$$

Definition 2.10. *Tableau Switching* ζ is a bijection between pairs

$$\zeta : \left\{ (T,U) \;\middle|\; \begin{matrix} T \in \mathrm{YT}(\pi/\mu, \mathbf{d}) \\ U \in \mathrm{YT}(\lambda/\pi, \mathbf{c}) \end{matrix} \text{ for some partition } \pi \text{ of } |\lambda| - |\mathbf{c}| \right\}$$
$$\leftrightarrow \left\{ (T',U') \;\middle|\; \begin{matrix} T' \in \mathrm{YT}(\sigma/\mu, \mathbf{c}) \\ U' \in \mathrm{YT}(\lambda/\sigma, \mathbf{d}) \end{matrix} \text{ for some partition } \sigma \text{ of } |\lambda| - |\mathbf{d}| \right\}$$

It is defined through the following process:

Algorithm 2.1 Tableau Switching Map[1]

```
Input: $k$, partitions $\mu \subset \pi \subset \lambda$ (partition inclusion), $\mathbf{a}, \mathbf{b}$; $A \in \mathrm{YT}(\lambda/\pi, \mathbf{a}), B \in \mathrm{YT}(\lambda/\mu, \mathbf{b})$
Output: $(A', B') = \zeta(B, A)$
 1: function $\zeta(A, B)$
 2:     Relabel integers in $A$ by adding $r$ to them
 3:     $\mathbf{e} \leftarrow (a_1, \dots, a_{m-r}, b_1, \dots, b_r)$
 4:     $\mathbf{f} \leftarrow (b_1, \dots, b_r, a_1, \dots, a_{m-r})$
 5:     $Y \leftarrow t_{r,m-r}(B \star A)$          ▷ $B \star A \in \mathrm{YT}(\lambda/\mu, \mathbf{f}) \implies Y \in \mathrm{YT}(\lambda/\mu, \mathbf{e})$
 6:     $C' \leftarrow \xi(C)$          ▷ $C' \in \mathrm{YT}(\lambda/\mu, (a_k, \dots, a_1, b_1, \dots, b_k))$
 7:     Decompose $A' \star B' = Y$ such that $A' \in \mathrm{YT}(\lambda/\sigma, (0, \dots, 0, b_1, \dots, b_k)), B'' \in \mathrm{YT}(\sigma/\mu, \mathbf{a}^*)$          ▷ $|\sigma| = |\mu| + |\mathbf{a}|$
 8:     Relabel integers in $B'$ by subtracting $r$ from them
 9:     return $(A', B')$
10: end function
```

Remark. Give two tableaux A, B of shape λ/π and π/μ, $B \star A$ gives a tableau f size λ/μ with entries obtained from each original tableau. One example is shown in Figure 2.

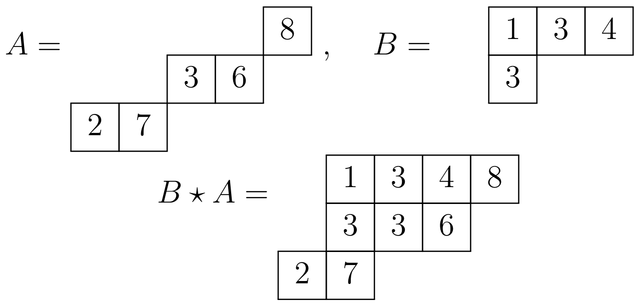


FIGURE 2. One example of $\star$ operation

Definition 2.11. The *Schützenberger Involution* $\xi = z_m$ is a bijection between

$$\mathrm{YT}(\lambda/\mu, \mathbf{a}) \leftrightarrow \mathrm{YT}(\lambda/\mu, \mathbf{a}^*)$$

where $\mathbf{a}^*$ denote the reverse of the sequence $\mathbf{a}$. Similarly ξ^N is the restriction onto non-skew tableaux i.e. $\mu = \emptyset$.

Definition 2.12. Reversal is another bijection between, defined by the conjugation of Schützenberger Involution with tableaux switching:

$$\chi = \zeta \circ \xi \circ \zeta : \mathrm{YT}(\lambda/\mu, \mathbf{a}) \leftrightarrow \mathrm{YT}(\lambda/\mu, \mathbf{a}^*)$$

Definition 2.13. *Fundamental Symmetry Maps* are a group of bijections between LR-tableaux:

$$\rho : \mathrm{LR}(\lambda/\mu, \nu) \leftrightarrow \mathrm{LR}(\lambda/\nu, \mu)/$$

The first fundamental symmetry map, ρ_1, is defined through tableaux switching: for $A \in \mathrm{LR}(\lambda/\mu, \nu)$, let $B = \mathrm{Can}(\mu)$. Then $\eta(B, A) = (A', B')$ where $A' = \phi(A) = \mathrm{Can}(\nu)$[2, §5.2]. We let $\rho_1(A) = B'$.

The second fundamental symmetry map, ρ_2, is defined as the composition of several maps.

2.3. **Equivalence Relations.**

Definition 2.14. For an input A, $\langle A \rangle$ denotes the bit-size of array A. A map $\delta : \mathcal{A} \to \mathcal{B}$ has *linear cost* if $\forall A \in \mathcal{A}, \delta(A)$ costs $O(\langle A \rangle)$ time complexity.

Definition 2.15 (Circuits)**.** We propose some ways to chain bijections, called circuits:

- Trivial circuit: given $\delta_1 : \mathcal{A} \to \mathcal{X}_1$, $\gamma : \mathcal{X}_1 \to \mathcal{X}_2$, and $\delta_2 : \mathcal{X}_2 \to \mathcal{B}$ where δ_1, δ_2 has linear cost, the map $\chi = \delta_2 \circ \gamma \circ \delta_1 : \mathcal{A} \to \mathcal{B}$ is called a trivial circuit and denoted by $I(\delta_1, \gamma, \delta_2)$.
- Sequential circuit: given $\gamma_1 : \mathcal{A} \to \mathcal{X}, \gamma_2 : \mathcal{X} \to \mathcal{B}$, the map $\chi = \gamma_2 \circ \gamma_1 : \mathcal{A} \to \mathcal{B}$ is a sequential circuit and denoted by $S(\gamma_1, \gamma_2)$.
- Parallel circuit: given $\delta_1 : \mathcal{A} \to \mathcal{X}_1 \times \mathcal{X}_2$, $\gamma_1 : \mathcal{X}_1 \to \mathcal{Y}_1$, $\gamma_2 : \mathcal{X}_2 \to \mathcal{Y}_2$, and $\delta_2 : \mathcal{Y}_1 \times \mathcal{Y}_2 \to \mathcal{B}$ where δ_1, δ_2 has linear cost, the map $\chi = \delta_2 \circ (\gamma_1 \otimes \gamma_2) \circ \delta_1 : \mathcal{A} \to \mathcal{B}$ is a parallel circuit and denoted by $P(\delta_1, \gamma_1, \gamma_2, \delta_2)$.

The definition should be able to generalize to finite products, but here 2 sets are enough given our bijections introduced above.

Definition 2.16. Given a bijection β, we can recursively define a β-based ps-circuit (parallel-sequential) $\daleth$ as follows:

- $\daleth = \delta$ where δ has linear cost.
- $\daleth = I(\delta_1, \beta, \delta_2)$
- $\daleth = S(\gamma_1, \gamma_2)$ where γ_1, γ_2 are β-based ps-circuits
- $\daleth = P(\delta_1, \gamma_1, \gamma_2, \delta_2)$ where γ_1, γ_2 are β-based ps-circuits

The β-cost of $\daleth$, denoted by $s(\daleth)$, is the number of times the map β is used.

Definition 2.17. A map α is linearly reducible to β (denoted as $\alpha \hookrightarrow \beta$) if there exist a finite β-based ps-circuit $\daleth$ which computes α. In this case we say that α can be computed in at most $s(\daleth)$ cost of β.

Definition 2.18. Maps α and β are *linearly equivalent* (denoted as $\alpha \sim \beta$) if α is linearly reducible to β and β and β is linearly reducible to α.

Lemma 2.1. *Suppose $\alpha_1 \hookrightarrow \alpha_2$ and $\alpha_2 \hookrightarrow \alpha_3$ then $\alpha_1 \hookrightarrow \alpha_3$. Moreover, if α_1 can be computed in at most $s1$ cost of α_2, and α_2 can be computed in at most s_2 cost of α_3, then α_1 can be computed in at most $(s_1 s_2)$ cost of α_3.*

Corollary 2.1. *Suppose $\alpha_1 \sim \alpha_2$ and $\alpha_2 \sim \alpha_3$ then $\alpha_1 \sim \alpha_3$.*

So linear equivalence is an equivalence relation.

Corollary 2.2 (Cycle Lemma). *Suppose $\alpha_1 \hookrightarrow \alpha_2 \hookrightarrow \alpha_3 \hookrightarrow \ldots \hookrightarrow \alpha_n \hookrightarrow \alpha_1$ then $\alpha_1 \sim \alpha_2 \sim \ldots \sim \alpha_n$.*

Remark. We should notice that the linear reduction $\daleth$ does not guarantee linear time reduction: each map can be computed in linear time with respect to the input size, but there are no restrictions on how the input size can change through the circuit. But as you may have expected, the maps in the circuit alters input sizes in a linear, bounded manner, so we can treat them as linear time reductions.

3. Linear Reductions

3.1. **Overview.** The idea is we want to show all the maps in Section 2.2 are linearly equivalent. Corollary 2.2 lets us achieve this through finding some overlapping cycles.

Lemma 3.1. *Such linear reduction cycles exist:*

- $\varphi \hookrightarrow \psi \hookrightarrow \phi \hookrightarrow \zeta^N \hookrightarrow \xi^N \hookrightarrow \varphi$.
- $\rho_1 \hookrightarrow \zeta^N \hookrightarrow \zeta \hookleftarrow \rho_1$ *and* $\rho_2 \hookrightarrow \xi^N \hookrightarrow \rho_2$.
- $\chi \hookrightarrow \xi^N \hookrightarrow \chi$.

Theorem 3.1. *$\varphi, \psi, \phi, \zeta, \zeta^N, \xi^N, \chi, \rho_1, \rho_2$ are linearly equivalent. Moreover, each of these maps can be computed in at most 36 times the cost of any other map.*

Proof. By Lemma 3.1, we can draw the following diagram. Figure 3 shows the structure of reduction. By Corollary 2.2, we can also work out the constant of the time cost in the next section. ■

$$\begin{tikzcd}
 & \psi \arrow[rr] & & \phi \arrow[dd] & \\
 & & & & \rho_1 \arrow[dl] \\
\varphi \arrow[uur] & & & \zeta^N \arrow[dl] \arrow[dr] & \\
 & & \xi^N \arrow[ull] \arrow[dll, bend right] \arrow[dr, bend left] & & \zeta \arrow[uu] \\
\chi \arrow[urr, bend right] & & & \rho_2 \arrow[ul, bend left] &
\end{tikzcd}$$

FIGURE 3. Reduction Diagram

Theorem 3.2. *For the eight maps as in Theorem 3.1, let k and m be defined as follows:*

	k	m
φ	$\max\{\ell(\mathbf{a}), \ell(\mathbf{b})\}$	$\max\left\{\sum_i a_i, \sum_j b_j\right\}$
$\psi, \phi, \zeta^N, \chi$	$\ell(\mathbf{a})$	λ_1
ζ	$\max\{\ell(\mathbf{a}), \ell(\mathbf{b})\}$	λ_1
ρ_1, ρ_2	$\ell(\nu)$	λ_1

where $\ell(\cdot)$ denotes the length of the partition/sequence.

Then the image of maps above can be computed at a cost of $O(k^3 \log m)$.

3.2. **Reductions.** We need to show the existence of these cycles mentioned in Lemma 3.1 exist.

Lemma 3.2. *Algorithm 3.1 shows $\varphi \hookrightarrow \psi$.*

Algorithm 3.1 $\varphi \hookrightarrow \psi$

```
Input: a, b; $V \in \mathbb{Z}_{\geq 0}^{k\times k}$ with row sum a and column sum b.
Output: $(B, A) = \varphi(V)$
 1: function RED(a, b, $V$)
 2:     $\pi \leftarrow (a_1 + \ldots + a_k, a_1 + \ldots + a_{k-1}, \ldots, a_1)$
 3:     $\sigma \leftarrow (a_1 + \ldots + a_{k-1}, a_1 + \ldots + a_{k-2}, \ldots, 0)$
 4:     $\rho \leftarrow (b_1 + \ldots + b_k, b_1 + \ldots + b_{k-1}, \ldots, b_1)$
 5:     $\tau \leftarrow (b_1 + \ldots + b_{k-1}, b_1 + \ldots + b_{k-2}, \ldots, 0)$
 6:     $V^{\updownarrow}_{i,j} \leftarrow V_{k+1-i,j}$                          ▷ reversing rows
 7:     $V'^{\updownarrow}_{i,j} \leftarrow V_{j,k+1-i}$          ▷ transposing and reversing columns
 8:     Let $Y \in \mathrm{YT}(\pi/\sigma, \mathbf{b})$ be the tableau with recording matrix $V^{\updownarrow}$
 9:     Let $X \in \mathrm{YT}(\rho/\tau, \mathbf{a})$ be the tableau with recording matrix $V'^{\updownarrow}$
10:     $B, A \leftarrow \psi(Y), \psi(X)$
11:     return $(B, A)$
12: end function
```

Remark. This is a parallel circuit where γ_1 refers to line 8, and γ_2 refers to line 9 of Algorithm 3.1. The correctness is verified through [4].

Lemma 3.3. *$\psi \hookrightarrow \phi$ through restrict the output of ϕ to the standard tableau i.e $\phi(A) = (B, C) \implies \psi(A) = B$.*

Lemma 3.4. *Algorithm 3.2 shows $\varphi \hookrightarrow \zeta^N$.*

Algorithm 3.2 $\varphi \hookrightarrow \zeta^N$

```
Input: $\mathbf{a}$, partitions $\lambda, \mu$, such that $\ell(\mathbf{a}), \ell(\lambda) \le k$; $A \in \mathrm{YT}(\lambda/\mu, \mathbf{a})$
Output: $(A', C') \in \mathrm{YT}(\sigma, =fa) \times \mathrm{LR}(\lambda/\mu, \sigma)$
 1: function $\textsc{Red}(\mathbf{a}, \lambda, \mu, A)$
 2:     $B \leftarrow \mathrm{Can}(\mu)$
 3:     $(A', B') \leftarrow \zeta^N(B, A)$
 4:     $\sigma \leftarrow$ shape of $A$
 5:     $C \leftarrow \mathrm{Can}(\sigma)$
 6:     $(B'', C') \leftarrow \zeta^N(C, B')$
 7:     return $(A', C')$
 8: end function
```

Remark. Algorithm 3.2 is a sequential circuit which uses ζ^N twice.

Lemma 3.5. *Algorithm 3.3 shows $\zeta \hookrightarrow \xi$. Consequently, $\zeta^N \hookrightarrow \xi^N$.*

Algorithm 3.3 $\zeta \hookrightarrow \xi$

```
Input: $k$, partitions $\mu \subset \pi \subset \lambda$ (partition inclusion), $\mathbf{a}, \mathbf{b}$; $A \in \mathrm{YT}(\lambda/\pi, \mathbf{a}), B \in \mathrm{YT}(\lambda/\mu, \mathbf{b})$
Output: $(A'', B'') = \zeta(B, A)$
 1: function $\textsc{Red}(\mu, \pi, \lambda, A, B)$
 2:     $B' \leftarrow \xi(B)$                         ▷ $B' \in \mathrm{YT}(\pi/\mu, \mathbf{b}^*)$ where $\mathbf{b}^* = (b_k, \dots, b_1)$
 3:     Relabel integers in $A$ by adding $k$ to them     ▷ ensure next step is well-defined
 4:     $C \leftarrow B' \star A$                       ▷ $C \in \mathrm{YT}(\lambda/\mu, (b_k, \dots, b_1, a_1, \dots, a_k))$
 5:     $C' \leftarrow \xi(C)$                          ▷ $C' \in \mathrm{YT}(\lambda/\mu, (a_k, \dots, a_1, b_1, \dots, b_k))$
 6:     Decompose $A' \star B'' = C$ such that $A' \in \mathrm{YT}(\lambda/\sigma, (0, \dots, 0, b_1, \dots, b_k)), B'' \in \mathrm{YT}(\sigma/\mu, \mathbf{a}^*)$      ▷ $|\sigma| = |\mu| + |\mathbf{a}|$
 7:     Relabel integers in $B''$ by subtracting $k$ from them
 8:     $A'' \leftarrow \xi(A')$                        ▷ $A'' \in \mathrm{YT}(\sigma/\mu, \eth)$
 9:     return $(A'', B'')$
10: end function
```

Remark. Algorithm 3.3 gives a sequential circuit which uses ξ three times.

Lemma 3.6. *Algorithm 3.4 shows $\xi^N \hookrightarrow \varphi$.*

This is a relative simple algorithm.

Algorithm 3.4 $\xi^N \hookrightarrow \varphi$

```
Input: $k$, $\lambda, \mathbf{a}$ such that $\ell(\lambda), \ell(\mathbf{a}) \leq k$; $A \in \mathrm{YT}(\lambda, \mathbf{a})$
Output: $(A \deg) = \zeta^N(A) \in \mathrm{YT}(\lambda, \mathbf{a}^*)$
  function Red($\mu, \pi, \lambda, A, B$)
      $V \leftarrow (v_{i,j})$ the recording matrix of $A$
      $U = (u_{i,j}) \leftarrow (v_{i,k+1-j})$
      $(A^\circ, B^\circ) \leftarrow \varphi(U)$
      return $A^\circ$
  end function
```

Remark. Algorithm 3.4 is a trivial circuit which uses φ only once.

This shows the existence of the first cycles. The rest is shown in detail in [3].

4. Notable Remarks

4.1. **Symmetry Maps.** Empirical evidence suggests that $\rho_1 = \rho_2$. In that case, we would have the following diagram. This can help reduce the constant 36 mentioned in Theorem 3.1.

$$\begin{tikzcd}
 & \psi \arrow[rr] & & \phi \arrow[dr] & & \\
\varphi \arrow[ur] & & & & \zeta^N \arrow[dll] \arrow[dr] & \\
 & & \xi^N \arrow[ull] \arrow[dll, bend right] \arrow[drr, bend left] & & & \zeta \arrow[dl] \\
\chi \arrow[urr, bend right] & & & & \rho_{1/2} \arrow[u] \arrow[ull, bend left] &
\end{tikzcd}$$

Figure 4. Reduction Diagram when $\rho_1 = \rho_2$

4.2. **Inversions.** We have not shown that the general Schützenberger involution ξ reduces to the other bijections appearing in this paper, while ξ^N and χ do. Proving that ξ is reducible to ξ^N remains an open problem.

References

[1] Georgia Benkart, Frank Sottile, and Jeffrey Stroomer. Tableau switching: Algorithms and applications. *Journal of Combinatorial Theory, Series A*, 76(1):11–43, 1996.

[2] William Fulton. *Young Tableaux: With Applications to Representation Theory and Geometry.* London Mathematical Society Student Texts. Cambridge University Press, 1996.

[3] Igor Pak and Ernesto Vallejo. Reductions of young tableau bijections, 2004.

[4] Glânffrwd P. Thomas. On a construction of schützenberger. *Discrete Mathematics*, 17(1):107–118, 1977.