Discrete Mathematics 51 (1984) 101–108
North-Holland

A SHORT HOOK-LENGTHS BIJECTION INSPIRED BY THE GREENE–NIJENHUIS–WILF PROOF

Doron ZEILBERGER

Department of Mathematics, Drexel University, Philadelphia, PA 19104, USA

Received 16 November 1982
Revised 31 October 1983

The celebrated Frame–Robinson–Thrall (Canad. J. Math. **6** (1954) 316–324) hook-lengths formula, counting the Young tableaux of a specified shape, is given a short bijective proof. This proof was obtained by translating the elegant Greene–Nijenhuis–Wilf proof (Adv. in Math. 31 (1979) 104–109) into bijective language.

0. Getting hooked

A Young tableau of shape $\lambda = (\lambda_1, \ldots, \lambda_m)$, $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_m > 0$, is an array $(a_{ij}: 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant \lambda_i)$ satisfying $a_{ij} < a_{i+1,j}$ and $a_{ij} < a_{i,j+1}$ (whenever applicable) such that every integer between 1 and $n(=\lambda_1 + \cdots + \lambda_m)$ appears exactly once among its n entries. For example

$$\begin{array}{lll} 1 & 2 & 4 \\ 3 & 6 & 10 \\ 5 & 7 & \\ 8 & 9 & \end{array}$$

is a Young tableau of shape $(3, 3, 2, 2)$.

The set of cells $\{(i, j) : 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant \lambda_i\}$ constitutes the *shape* of the tableau, $S(\lambda)$, and for every cell (i, j) in $S(\lambda)$ we define its *hook* H_{ij} by $H_{ij} = \{(\alpha, \beta) \in S(\lambda) : \alpha = i \text{ and } \beta \geqslant j \text{ or } \alpha \geqslant i \text{ and } \beta = j\}$. The number of cells in H_{ij} is denoted by h_{ij}.

Frame, Thrall and Robinson [1] proved that the number of Young tableaux of shape λ, f_λ, is given by

$$f_\lambda = n! \Big/ \prod_{(i,j) \in S(\lambda)} h_{ij} \tag{1}$$

For example, if $\lambda = (2, 2)$, then $n = 2 + 2 = 4$ and

$$\begin{aligned} &H_{11} = \{(1, 1), (1, 2), (2, 1)\}, && h_{11} = 3, \\ &H_{12} = \{(1, 2), (2, 2)\}, && h_{12} = 2, \\ &H_{21} = \{(2, 1), (2, 2)\}, && h_{21} = 2, \\ &H_{22} = \{(2, 2)\}, && h_{22} = 1, \end{aligned}$$

and $f_\lambda = 4!/(3\cdot 2\cdot 2\cdot 1) = 24/12 = 2$ and indeed there are 2 Young tableaux of shape $(2,2)$: $\substack{1\,2\\3\,4}$ and $\substack{1\,3\\2\,4}$.

An excellent exposition of what was known about Young tableaux until 1972 is given in [5]. Knuth [5, p. 63] comments: "Since the hook-lengths formula is such a simple result, it deserves a simple proof... All known proofs of this formula are based on an uninspiring induction argument which does not really explain why the theorem is true (since it does not really use the properties of hooks)."

Since these words were written, Greene, Nijenhuis and Wilf [4] used very inspired induction in a cute proof they gave which uses the properties of hooks in a very essential way. A bijective proof of $f_\lambda \prod h_{ij} = n!$ is given in [2]. However, the mapping involved does not portray the nice row-column symmetry of the hooks, as it is heavily lopsided toward the columns. Furthermore, the proof that the mapping is indeed a bijection is much longer than should be desired.

In the present paper we give a *bijective proof* of the hook-lengths formula which possesses all the virtues of the GNW probabilistic-inductive proof. This 'conservation of elegance' is by no means a coincidence, as our proof is a direct 'bijectation' of the GNW proof. The notion of 'bijectation' or combinatorization of manipulative proofs was first made explicit in the innovative work of Garsia and Milne [3] and was further made use of in the works of Remmel [6] who gave the first bijective proof of the hook-lengths formula [7]. The present author has extended the bijectation method to inductive proofs and hopes to present it elsewhere.

1. The theorem

Definition 1. A pointer tableau of shape λ is an assignment of pointers to every cell of $S(\lambda)$ such that every cell points at some member of its hook. More formally, it is an array

$$\{P(i,j) : 1 \leq i \leq m,\ 1 \leq j \leq \lambda_i,\ P(i,j) \in H_{ij}\}.$$

Definition 2. Let $S'(\lambda)$ be the set of non-corner cells in $S(\lambda)$, i.e., those (i,j) for which $h_{ij} > 1$.

Definition 3. A pointer tableau is *strict* if for every $(i,j) \in S'(\lambda)$ $P(i,j) \neq (i,j)$, i.e., only corner cells point at themselves.

The set of pointer tableaux and strict pointer tableaux of shape λ are denoted by $\mathcal{P}(\lambda)$ and $\mathcal{P}_s(\lambda)$ respectively. The set of Young tableaux of shape λ is denoted by $\mathcal{T}(\lambda)$ and S_n is the set of permutations on $\{1,\ldots,n\}$.

Theorem. *The mapping*

$$\tau(\lambda) : S_n \times \bigtimes_{\mu \subseteq \lambda} \mathcal{P}_s(\mu) \to \mathcal{T}(\lambda) \times \mathcal{P}(\lambda) \times \bigtimes_{\mu \subseteq \lambda} \mathcal{P}_s(\mu)$$

defined in Section 2.1 is a bijection and the mapping

$$\sigma(\lambda):\mathcal{T}(\lambda)\times\mathcal{P}(\lambda)\times \mathop{\mathsf{X}}_{\mu\subseteq\lambda}\mathcal{P}_s(\mu)\rightarrow S_n\times\mathop{\mathsf{X}}_{\mu\subseteq\lambda}\mathcal{P}_s(\mu)$$

defined in Section 2.2 is its inverse.

The hook-lengths formula is an immediate corollary since the theorem implies that

$$\left|S_n\times\mathop{\mathsf{X}}_{\mu\subseteq\lambda}\mathcal{P}_s(\mu)\right|=\left|\mathcal{T}(\lambda)\times\mathcal{P}(\lambda)\times\mathop{\mathsf{X}}_{\mu\subseteq\lambda}\mathcal{P}_s(\mu)\right|$$

and thus

$$|S_n|\left|\mathop{\mathsf{X}}_{\mu\subseteq\lambda}\mathcal{P}_s(\mu)\right|=|\mathcal{T}(\lambda)|\,|\mathcal{P}(\lambda)|\left|\mathop{\mathsf{X}}_{\mu\subseteq\lambda}\mathcal{P}_s(\mu)\right|,$$

and so

$$|\mathcal{T}(\lambda)|=\frac{|S_n|\left|\mathop{\mathsf{X}}_{\mu\subseteq\lambda}\mathcal{P}_s(\mu)\right|}{|\mathcal{P}(\lambda)|\left|\mathop{\mathsf{X}}_{\mu\subseteq\lambda}\mathcal{P}_s(\mu)\right|}=\frac{|S_n|}{|\mathcal{P}(\lambda)|}=n!/\prod_{(i,j)\in S(\lambda)}h_{ij}.$$

The proof of the theorem will consist in presenting the mappings $\pi(\lambda)$ and $\sigma(\lambda)$. The proof that the algorithms describing $\pi(\lambda)$ and $\sigma(\lambda)$ do what they claim to do is immediate while the proof that $\pi(\lambda)$ and $\sigma(\lambda)$ are inverses of each other follows from the fact that $\sigma(\lambda)$'s Step i ($1\leq i\leq 6$) undoes $\pi(\lambda)$'s Step 7-i and vice versa, and from the inductive hypothesis.

For $(a,b)\in S(\lambda)$ let $H^*_{ab}=H_{ab}\setminus(a,b)$. Let (α,β) be a corner cell in $S(\lambda)$ and let (a,b) be any cell in $S(\lambda)$ for which $a\leq\alpha$ and $b\leq\beta$ the bijection $f^{ab}_{\alpha\beta}:H^*_{\alpha\beta}\cup H^*_{ab}\rightarrow H^*_{ab}$ (establishing $(h_{\alpha\beta}-1)+(h_{ab}-1)=h_{ab}-1$) is defined by

$$f^{ab}_{\alpha\beta}:\quad H^*_{\alpha\beta}\begin{cases}(a,y)\rightarrow(a,y)\\(x,\beta)\rightarrow(x,b)\end{cases}\qquad H^*_{\alpha b}\begin{cases}(\alpha,y)\rightarrow(a,y)\\(x,b)\rightarrow(x,b)\end{cases}$$

The inverse of $f^{ab}_{\alpha\beta}$ is easily seen to be given by

$$(f^{ab}_{\alpha\beta})^{-1}:\quad (a,y)\begin{cases}\xrightarrow{y>\beta}(a,y)\in H^*_{\alpha\beta}\\ \xrightarrow{y\leq\beta}(\alpha,y)\in H^*_{\alpha b}\end{cases},\qquad (x,b)\begin{cases}\xrightarrow{x\leq\alpha}(x,\beta)\in H^*_{ab}\\ \xrightarrow{x>\alpha}(x,b)\in H^*_{\alpha b}\end{cases}.$$

The knowledge of $f^{ab}_{\alpha\beta}$ and its inverse is *crucial* for the execution of algorithms $\pi(\lambda)$ and $\sigma(\lambda)$.

2. The bijection and its inverse

Warning. Familiarize yourself with the $f^{ab}_{\alpha\beta}$ defined at the end of Section 1.

Notation. (1) The *reduced form* of the permutation $(a_1, \ldots, a_k)$ is $(b_1, \ldots, b_k)$, where $\{b_1, b_2, \ldots, b_k\} = \{1, 2, \ldots, k\}$ and $a_i < a_j$ iff $b_i < b_j$; e.g., the reduced form of (5, 7, 8, 2) is (2, 3, 4, 1); red(9, 1, 2, 8) = (4, 1, 2, 3).

(2) The (i, j) entry of an array A is denoted by $A(i, j)$.

2.1. *Algorithm* $\pi(\lambda)$

Input. $(P_\mu)_{\mu \subseteq \lambda}$ and $x_n = (m_1, \ldots, m_n)$. P_μ are strict pointer tableaux of shape μ, for every $\mu \subseteq \lambda$; x_n is a permutation of $\{1, \ldots, n\}$.

Output. $(Q_\mu)_{\mu \subseteq \lambda}$, K_λ and T_λ; Q_μ is a strict pointer tableau of shape μ, for every $\mu \subseteq \lambda$; K_λ is a pointer tableau of shape λ and T_λ is a Young tableau of shape λ.

Step 1. [Locate beginning of trip using m_1] Let (a, b) be the m_1th cell of $S(\lambda)$ obtained by scanning it as in reading English, i.e., $m_1 = \lambda_1 + \cdots + \lambda_{a-1} + b$. Let x_{n-1} be the reduced form of $(m_2, \ldots, m_n)$.

Step 2. [Find end of trip] Starting at (a, b) follow the pointers of P_λ getting a path $(a, b) = (a_1, b_1) \to (a_2, b_2) \to \cdots \to (a_m, b_m) = (\alpha, \beta)$, where (α, β) is a corner cell. (i.e., $(a_i, b_i) = P_\lambda(a_{i-1}, b_{i-1})$, $i = 2, \ldots, m$, thus *either* $a_i = a_{i-1}$ and $b_i > b_{i-1}$ *or* $a_i > a_{i-1}$ and $b_i = b_{i-1}$)

Step 3. [Apply recursion] Let $\bar\lambda$ be the shape λ with the cell (α, β) deleted. Apply $\pi(\bar\lambda)$ to $(P_\mu)_{\mu \subseteq \bar\lambda}$ and x_{n-1} to get strict pointer tableaux $(Q_\mu)_{\mu \subseteq \bar\lambda}$, a pointer tableau $K_{\bar\lambda}$ and a young tableau $T_{\bar\lambda}$ of shape $\bar\lambda$.

Step 4. [Get $(Q_\mu)_{\mu \subseteq \lambda}$; initialize K_λ and Q_λ] Keep $(Q_\mu)_{\mu \subseteq \bar\lambda}$ which you got in Step 3; for $\bar\lambda \not\supseteq \mu \subseteq \lambda$ set $Q_\mu \leftarrow P_\mu$. Set $K_\lambda(x, y) = K_{\bar\lambda}(x, y)$ for $(x, y) \neq (\alpha, \beta)$ and $K_\lambda(\alpha, \beta) = (\alpha, \beta)$.

Step 5. [Update Q_λ and K_λ] For $i = 1, \ldots, m-1$ do

(a) [Find cell] If $a_{i+1} = a_i$ then cell $\leftarrow (\alpha, b_i)$ if $b_{i+1} = b_i$ then cell $\leftarrow (a_i, \beta)$.

(b) [Update $Q_\lambda(a_i, b_i)$, Q_λ(cell), K_λ(cell)]

$$Q_\lambda(a_i, b_i) \leftarrow f_{\alpha\beta}^{a_i b_i}(Q_\lambda(\text{cell})),$$

$$Q_\lambda(\text{cell}) \leftarrow \begin{cases} K_\lambda(\text{cell}) & \text{if } K_\lambda(\text{cell}) \neq \text{cell}, \\ (\alpha, \beta) & \text{if } K_\lambda(\text{cell}) = \text{cell}, \end{cases}$$

$$K_\lambda(\text{cell}) \leftarrow (\alpha, \beta).$$

Step 6. [Get T_λ] Let T_λ be the Young tableau obtained from $T_{\bar\lambda}$ by adjoining the cell (α, β) filled with 'n'.

2.2. *Algorithm* $\sigma(\lambda)$

Input. $(Q_\mu)_{\mu \subseteq \lambda}$, K_λ and T_λ; Q_μ is a strict pointer tableau of shape μ, for every $\mu \subseteq \lambda$; K_λ is a pointer tableau of shape λ and T_λ is a Young tableau of shape λ.

Output. $(P_\mu)_{\mu \subseteq \lambda}$ and x_n; P_μ is a strict pointer tableau of shape μ, for every $\mu \subseteq \lambda$; $x_n = (m_1, \ldots, m_n)$ is a permutation of $\{1, \ldots, n\}$.

Step 1. [Locate corner cell (α, β) and get $T_{\bar\lambda}$] Let $T_{\bar\lambda}$, of shape $\bar\lambda$, be the Young tableau obtained from T_λ by deleting 'n'. Let (α, β) be the corner cell which have been thus removed (i.e., $(\alpha, \beta) = \lambda \setminus \bar\lambda$).

Step 2. [Update Q_λ and K_λ]

(a) [Locate cells in the 'antihook' of (α, β) which point at (α, β) in K_λ] Let $(\alpha, d_1), \ldots, (\alpha, d_s) = (\alpha, \beta)$, and $(c_1, \beta), \ldots, (c_r, \beta) = (\alpha, \beta)$ be the cells (x, y) for which $K_\lambda(x, y) = (\alpha, \beta)$.

(b) [Initialize i and j] $i \leftarrow 1$, $j \leftarrow 1$.

(c) [Find cell] Let cell be such that

$$(f_{\alpha\beta}^{c_i d_j})^{-1}(Q_\lambda(c_i, d_j)) \in H^*_{\text{cell}} \quad \text{(i.e., cell} = (c_i, \beta) \text{ or } (\alpha, d_j)\text{)}.$$

(d) [Update $K_\lambda(\text{cell})$, $Q_\lambda(\text{cell})$]

$$K_\lambda(\text{cell}) \leftarrow \begin{cases} Q_\lambda(\text{cell}) & \text{if } Q_\lambda(\text{cell}) \neq (\alpha, \beta), \\ \text{cell} & \text{if } Q_\lambda(\text{cell}) = (\alpha, \beta). \end{cases}$$

$$Q_\lambda(\text{cell}) \leftarrow (f_{\alpha\beta}^{c_i d_j})^{-1}(Q_\lambda(c_i, d_j)).$$

(e) [Update $Q_\lambda(c_i, d_j)$, update (i, j)] If cell $= (c_i, \beta)$ then $Q_\lambda(c_i, d_j) \leftarrow (c_{i+1}, d_j)$ and $i \leftarrow i+1$. If cell $= (\alpha, d_j)$ then $Q_\lambda(c_i, d_j) \leftarrow (c_i, d_{j+1})$ and $j \leftarrow j+1$.

(f) [done?] If $(c_i, d_j) \neq (\alpha, \beta)$ (i.e. $i < s$ or $j < r$) go to Step 2(c).

Step 3. [Get P_μ, $\mu \not\subseteq \bar\lambda$, $K_{\bar\lambda}$] For $\mu \subseteq \lambda$ and $\mu \not\subseteq \bar\lambda$ set $P_\mu \leftarrow Q_\mu$; set $K_{\bar\lambda}(x, y) \leftarrow K_\lambda(x, y)$, $(x, y) \in S(\bar\lambda)$.

Step 4. [Apply recursion] Apply $\sigma(\bar\lambda)$ with $(Q_\mu)_{\mu \subseteq \bar\lambda}$, $T_{\bar\lambda}$ and $K_{\bar\lambda}$ to get $(P_\mu)_{\mu \subseteq \bar\lambda}$ and x_{n-1}.

Step 5. [Find beginning of trip] Retrieve (c_1, d_1) from Step 2(a), call it (a, b), that is $(a, b) \leftarrow (c_1, d_1)$.

Step 6. [Find m_1 using beginning of trip] Let m_1 be such that (a, b) is the m_1th cell encountered when 'reading $S(\lambda)$ in English' (i.e., $m_1 = \lambda_1 + \cdots + \lambda_{a-1} + b$). Let x_n be the permutation $(m_1, m_2, \ldots, m_n)$ where $(m_2, \ldots, m_n)$ is such that its reduced form is x_{n-1}.

3. Example

Due to the enormous size of the input and output and to the recursive nature of the algorithm, it is impossible to present a complete worked out example of a non-trivial size. We will thus confine ourselves to an example of $\pi(\lambda)$ where we arbitrarily prescribe the outcome of the recursive Step 3.

Let $\lambda = (5, 5, 5, 5, 3) = (5^4, 3)$; then $n = 23$ and we want to apply $\pi(\lambda)$ to

$$P_{(5^4,3)} = \begin{matrix} * & (1,4) & * & (3,4) & (2,5) \\ * & * & * & * & * \\ * & * & * & (3,5) & (4,5) \\ * & (4,4) & * & (4,5) & (4,5) \\ * & * & * & & \end{matrix}$$

$(P_t)_{\mu\subseteq(5^4,3)}$ and $\bar{x}_n=(2,m_2,\dots,m_n)$, where the content of the cells filled with '*' is immaterial and is not going to change throughout the execution of the algorithm.

Step 1. $(a,b)=(1,2)$.

Step 2. We get the trip $(1,2)\to(1,4)\to(3,4)\to(3,5)\to(4,5)$ so $(\alpha,\beta)=(4,5)$ and $m=5$.

Step 3. $\bar{\lambda}=(5,5,5,4,3)=(5^3,4,3)$. Assume, for the sake of example, that $\pi(\bar{\lambda})$ yielded $(Q_\mu)_{\mu\subseteq\bar{\lambda}}$ and

$$K_{\bar{\lambda}}=K_{(5^3,4,3)}=\begin{matrix} * & * & * & * & (2,5)\\ * & * & * & * & *\\ * & * & * & * & (3,5)\\ * & (4,3) & * & (4,4) & *\\ * & * & * & & \end{matrix}\quad\text{and}\quad T_{\bar{\lambda}}=\begin{matrix} * & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & \\ * & * & * & & \end{matrix},$$

where again, the content of the cells filled with '*' is immaterial and unchanged throughout the algorithm.

Step 4. We get our (Q_μ), $\mu\subseteq(5^3,4,3)$ and set $Q_{(5^4,2)}\leftarrow P_{(5^4,2)}$ etc. and initialize $Q_{(5^4,3)}\leftarrow P_{(5^4,3)}$.

Step 5. $i=1$: $(1,2)\to(1,4)$, $a_2=a_1$ so cell $\leftarrow(4,2)$,

$$\begin{aligned}&Q_{(5^4,3)}(1,2)\leftarrow f^{12}_{45}(4,4)=(1,4),\\&Q_{(5^4,3)}(4,2)\leftarrow(4,3),\\&K_{(5^4,3)}(4,2)\leftarrow(4,5).\end{aligned}$$

$i=2$: $(1,4)\to(3,4)$, $b_2=b_3$ so cell $\leftarrow(1,5)$,

$$\begin{aligned}&Q_{(5^4,3)}(1,4)\leftarrow f^{14}_{45}(2,5)=(2,4),\\&Q_{(5^4,3)}(1,5)\leftarrow(2,5),\\&K_{(5^4,3)}(1,5)\leftarrow(4,5).\end{aligned}$$

$i=3$: $(3,4)\to(3,5)$, cell $\leftarrow(4.4)$,

$$\begin{aligned}&Q_{(5^4,3)}(3,4)\leftarrow f^{34}_{45}(4,5)=(3,5),\\&Q_{(5^4,3)}(4,4)\leftarrow(4,5)\quad(\text{since } K_{(5^4,3)}(4,4)=(4,4)\text{ a self pointer}),\\&K_{(5^4,3)}(4,4)\leftarrow(4,5).\end{aligned}$$

$i=4$: $(3,5)\to(4,5)$, cell $\leftarrow(3,5)$,

$$\begin{aligned}&Q_{(5^4,3)}(3,5)\leftarrow f^{35}_{45}(Q_{(5^4,3)}(3,5)),\\&Q_{(5^4,3)}(3,5)\leftarrow(4,5)\quad[=(\alpha,\beta)],\text{ since,}\\&K_{(5^4,3)}(3,5)=(3,5)\quad\text{a self pointer.}\\&K_{(5^4,3)}\leftarrow(4,5).\end{aligned}$$

Thus

$$Q_{(5^4,3)} = \begin{array}{ccccc} * & (1,4) & * & (2,4) & (2,5) \\ * & * & * & * & * \\ * & * & * & (3,5) & (4,5) \\ * & (4,3) & * & (4,5) & (4,5) \\ * & * & * & & \end{array},$$

$$(K_{(5^4,3)} = \begin{array}{ccccc} * & * & * & * & (4,5) \\ * & * & * & * & * \\ * & * & * & * & (4,5. \\ * & (4,5) & * & (4,5) & (4,5) \\ * & * & * & & \end{array}$$

Step 6.

$$T_{(5^4,3)} = \begin{array}{ccccc} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & 23 \\ * & * & & & \end{array}.$$

We urge the reader to apply $\sigma(5^4, 3)$ to the above output and verify that one gets the above input back.

4. The purist's objection and our rebuttal

The purist would object that our proof of $n! = f_\lambda \prod h_{ij}$ was not purely bijective and that what we really proved was the fact $n!\, A_\lambda = f_\lambda \prod h_{ij} A_\lambda$, for some number A_λ. To get $n! = f_\lambda \prod h_{ij}$ we had to go through the algebraic (and hence manipulative) act of cancelling A_λ out.

To this we retort that the hook-lengths formula states that $f_\lambda = n!/\prod h_{ij}$ and to get this from $n! = f_\lambda \prod h_{ij}$ also requires an algebraic manipulation. However, even if the original statement of the hook-lengths formula would have been $n! = f_\lambda \prod h_{ij}$, there is nothing wrong in proving $S_n \times \mathsf{X}_{\mu \subseteq \lambda} \mathcal{P}_s(\mu) \leftrightarrow \mathcal{T}(\lambda) \times \mathcal{P}(\lambda) \times \mathsf{X}_{\mu \subseteq \lambda} \mathcal{P}_s(\mu)$ rather than $S_n \leftrightarrow \mathcal{T}(\lambda) \times \mathcal{P}(\lambda)$, as long as the former is more elegant and gives more insight into the structure of Young tableaux and the properties of hooks.

Indeed, it may happen that two sets A and B have the same cardinality without any apparent bijective reason. Then it often happens that there exists another set C such that $A \times C$ and $B \times C$ have a very natural bijection. The introduction of the 'catalizator' C not only facilitates a bijective proof that $|A| = |B|$ but often gives insight into the structure of both A and B. This point is best illustrated by the following example.

The *green* couples 1, 2, 3 play hockey with the *red* couples 1, 2, 3. The positions are as follows:

Green

	Goalie: Mr. 1	
Left defense: Mrs. 1		Right defense: Mr. 2
Left wing: Mrs. 2	Center: Mr. 3	Right wing: Mrs. 3

Red

	Goalie: Mrs. 2′	
Left defense: Mr. 2′		Right defense: Mrs. 3′
Left wing: Mr. 3′	Center: Mr. 1′	Right wing: Mrs. 1′

If you were requested to give a bijective proof that the number of green couples equals the number of red couples, you would find *no* natural bijection $\{1, 2, 3\} \leftrightarrow \{1', 2', 3'\}$. The *natural* bijection is

$$\pi : \{\text{Mr., Mrs.}\} \times \{1, 2, 3\} \to \{\text{Mr., Mrs.}\} \times \{1', 2', 3'\}$$

given by $\pi(a) =$ the person in the red team having the same position as a. Thus $\pi(\text{Mr. } 1) = \text{Mrs. } 2'$; $\pi(\text{Mrs. } 1) = \text{Mr. } 2'$; etc.

Acknowledgment

Many thanks are due to ref. Hank Fiore for teaching me the rules of Hockey.

References

[1] J.S. Frame, G.D. Robinson and R.M. Thrall, The hook graphs of the symmetric group, Canadian J. Math. 6 (1954) 316–324.

[2] D. Franzblau and D. Zeilberger, A bijection proof of the hook-lengths formula, J. Algorithms.

[3] A.M. Garsia and S.C. Milne, A Rogers–Ramanujan bijection, J. Combin. Theory Ser. A 31 (1981) 289–339.

[4] C. Greene, A. Nijenhuis and H.S. Wilf, A probabilistic proof of a formula for the number of Young tableaux of a given shape, Adv. in Math. 31 (1979) 104–109.

[5] D.E. Knuth, The Art of Computer Programming, Vol. 3, Sorting and Searching (Addison-Wesley, Reading, MA, 1973).

[6] J.B. Remmel, Bijective proofs of some classical partition identities, J. Combin. Theory, in press.

[7] J.B. Remmel, Bijective proofs of formulae for the number of standard Young tableaux, UCSD preprint.