4/1

The Robinson-Schensted Algorithm

Recall f^λ = # SYTs of sh. λ. We will prove following identity.

Thm $\displaystyle\sum_{\lambda \vdash n} (f^\lambda)^2 = n!$ for all $n \geq 1$.

e.g. $n=2 \Rightarrow (f^{(2)})^2 + (f^{(1,1)})^2 = 1^2 + 1^2 = 2!$ ✓

$n=3 \Rightarrow (f^{(3)})^2 + (f^{(2,1)})^2 + (f^{(1,1,1)})^2 = 1^2 + 2^2 + 1^2 = 6 = 3!$ ✓

May seem like a strange formula, but has algebraic meaning.
We will explain a bijective pf. of this thm., using a very important procedure called Robinson-Schensted Algorithm.

Observe that

$$\sum_{\lambda \vdash n} (f^\lambda)^2 = \# \left\{ \begin{array}{l} \text{pairs } (P,Q) \text{ of SYTs w/ } n \text{ boxes,} \\ \text{s.t. } sh(P) = sh(Q) \end{array} \right\}$$

and of course

$$n! = \# \text{ permutations in } S_n$$

so the thm. would follow from a bijection

$$S_n \longrightarrow \{ \text{pairs } (P,Q) \text{ of SYTs w/ } sh(P) = sh(Q) \vdash n \}$$

The Robinson-Schensted Algorithm is such a bijection.

The main "loop" of the RS algorithm involves insertion: we have a tableau and we want to put a new # in it.

e.g.
$$\begin{array}{|c|c|c|c|} \hline 1 & 3 & 6 & 9 \\ \hline 2 & 8 & 10 \\ \cline{1-3} 4 \\ \cline{1-1} 7 \\ \cline{1-1} \end{array} \longleftarrow 5 \text{ insert}$$

Note: the "tableau" here is like an SYT in that #'s increase down rows/cols, but #s are not $1, 2, \ldots, n$. That's ok.

How do we carry out the insertion? Well we start by trying to put the # we're inserting (call it i) in the top row:

- If i is bigger than all #'s in 1st row, put it at end,
- otherwise, put i where smallest # bigger than i (call it j) is, and <u>bump</u> this j by inserting it into the next row.

e.g.
$$\begin{array}{l}1\ 3\ 6\ 9 \leftarrow 5\\ 2\ 8\ 10\\ 4\\ 7\end{array} \overset{5 \text{ bumps } 6}{\rightsquigarrow} \begin{array}{l}1\ 3\ 5\ 9\\ 2\ 8\ 10 \leftarrow 6\\ 4\\ 7\end{array} \overset{6 \text{ bumps } 8}{\rightsquigarrow} \begin{array}{l}1\ 3\ 5\ 9\\ 2\ 6\ 10\\ 4 \leftarrow 8\\ 7\end{array} \rightsquigarrow \boxed{\begin{array}{l}1\ 3\ 5\ 9\\ 2\ 6\ 10\\ 4\ 8\\ 7\end{array}}$$

As depicted above, we keep doing the same procedure of bumping until we reach a row where the # we're inserting is biggest. The result is the insertion of the # into the tableau, and <u>Exercise</u>: it produces a new tableau.

The RS algorithm is built out of these insertions. We start with permutation $\sigma = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ \sigma_1 & \sigma_2 & \sigma_3 & \cdots & \sigma_n \end{pmatrix} \in S_n$. We want to produce two SYTs, P and Q, of the same shape. The tableau P is called the <u>insertion</u> <u>tableau</u> and is the result of inserting σ_1, then σ_2, then $\sigma_3, \cdots$ (starting from $\emptyset$ empty tableau)

<u>e.g.</u> $\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 5 & 2 & 3 & 6 & 4 & 1 & 7 \end{pmatrix}$

$$\emptyset,\ \boxed{5},\ \begin{array}{l}2\\5\end{array},\ \begin{array}{l}2\ 3\\5\end{array},\ \begin{array}{l}2\ 3\ 6\\5\end{array},\ \begin{array}{l}2\ 3\ 4\\5\ 6\end{array},\ \begin{array}{l}1\ 3\ 4\\2\ 6\\5\end{array},\ \begin{array}{l}1\ 3\ 4\ 7\\2\ 6\\5\end{array} = P$$

Meanwhile, Q is the <u>recording</u> <u>tableau</u> and keeps track of the order in which boxes were added in insertion process:

<u>e.g.</u>
$$\emptyset,\ \boxed{1},\ \begin{array}{l}1\\2\end{array},\ \begin{array}{l}1\ 3\\2\end{array},\ \begin{array}{l}1\ 3\ 4\\2\end{array},\ \begin{array}{l}1\ 3\ 4\\2\ 5\end{array},\ \begin{array}{l}1\ 3\ 4\\2\ 5\\6\end{array},\ \begin{array}{l}1\ 3\ 4\ 7\\2\ 5\\6\end{array} = Q$$

4/4

By construction, P and Q have the same shape. So we get a map $S_n \xrightarrow{RS} \{(P,Q) : sh(P) = sh(Q) \vdash n\}$

Thm This map $\sigma \xmapsto{RS} (P,Q)$ is a bijection.

Pf: As w/ Hillman-Grassl, goal is to show we can locally undo steps. In other words, we can describe inverse $(P,Q) \xmapsto{RS^{-1}} \sigma$. Here is how that works. Suppose we are given (P,Q):

e.g. $P = \begin{array}{ccc} 1 & 3 & 4 \\ 2 & 6 & \\ 5 & & \end{array}$ $\qquad Q = \begin{array}{ccc} 1 & 3 & 4 \\ 2 & 5 & \\ 6 & & \end{array}$

The location of the biggest #, n, in Q tells us the # in P that was the termination of the last bumping sequence. Then we can "reverse bump/insert" this entry out of P:

- if it is in row 1, simply remove it,
- otherwise, have it replace the ~~smallest~~ largest # less than it in the row above, and bump that # out and repeat.

e.g. $\begin{array}{ccc} 1 & 3 & 4 \\ 2 & 6 & \\ \boxed{5} & & \end{array} \overset{5 \text{ bumps } 2}{\rightsquigarrow} \begin{array}{cccc} & 1 & 3 & 4 \\ \boxed{2} & 5 & 6 & \end{array} \overset{2 \text{ bumps } 1}{\rightsquigarrow} \begin{array}{cccc} 2 & 3 & 4 & \boxed{1} \\ 5 & 6 & & \end{array} \longrightarrow$ removed

rev. insert this $\rightarrow \boxed{5}$

Then write down $\sigma_n :=$ # removed from rev. insertion. And remove n from Q, and repeat same steps but now with box containing n-1 in Q. In this way, we build up sequence $\sigma_n, \sigma_{n-1}, \ldots, \sigma_1$ and then $\sigma = \sigma_1 \sigma_2 \cdots \sigma_n$ is our desired permutation.

It is easy to see that this is the inverse, b.c. reverse insertion "locally" inverses insertion (again there are a few things to check... I leave to you as Exercise). ∎

The Robinson-Schensted-Knuth Algorithm

The RSK algorithm is an extension of RS alg. to semistandard (as opposed to standard) tableaux. Again we have a motivational formula:

Thm (Cauchy identity)

$$\sum_{\lambda} s_\lambda(x_1, x_2, \ldots) s_\lambda(y_1, y_2, \ldots) = \prod_{i,j \geq 1}^{\infty} \frac{1}{(1 - x_i y_j)}$$

Before we start the proof, a few remarks about this identity:

- the sum is over all partitions λ (of all sizes)
- there are two infinite sets of variables $\vec{x} = \{x_1, x_2, \ldots\}$ and $\vec{y} = \{y_1, y_2, \ldots\}$
- it is an identity in $\mathbb{C}[[x_1, x_2, \ldots, y_1, y_2, \ldots]]$

The Cauchy identity is again very important result in sym. fn. theory.

By standard limit argument we've seen before, it suffices to prove a "finite" version for all $n \geq 1$:

$$\sum_{\lambda : \ell(\lambda) \leq n} s_\lambda(x_1, x_2, \ldots, x_n) s_\lambda(y_1, y_2, \ldots, y_n) = \prod_{i,j=1}^{n} \frac{1}{1 - x_i y_j}$$

Here we use only finitely many variables.

We want to give a bijective pf. of Cauchy identity so let's interpret the coefficient of $\vec{x}^{\alpha} \vec{y}^{\beta}$ for $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, $\beta = (\beta_1, \beta_2, \ldots, \beta_n)$ on LHS + RHS.

On LHS,

$$\text{coeff. of } \vec{x}^{\alpha}\vec{y}^{\beta} = \#\left\{(P,Q) : \begin{array}{l} P \text{ and } Q \text{ are SSYT w/} \\ sh(P) = sh(Q) \text{ and} \\ con(P) = \alpha \text{ and } con(Q) = \beta \end{array}\right\}$$

matrices w/ entries in $\mathbb{N} = \{0,1,2,\dots\}$

What about RHS? For this we will use $n\times n$ $\mathbb{N}$-matrices M:

- for $M=(m_{ij})$, let $\mathrm{row}_i(M) = \sum_j m_{ij}$ be sum of i^{th} row, and let $\mathrm{col}_j(M) = \sum_i m_{ij}$ be sum of j^{th} col.

Prop. Coeff. of $\vec{x}^{\alpha}\vec{y}^{\beta}$ in $\prod_{i,j\geq 1} \frac{1}{1-x_iy_j} = \#\left\{ \begin{array}{l} n\times n\ \mathbb{N}\text{-matrices } M \\ \text{w/ } \beta = (\mathrm{row}_1(M), \mathrm{row}_2(M), \dots), \\ \alpha = (\mathrm{col}_1(M), \mathrm{col}_2(M), \dots) \end{array} \right\}$

Pf. Associate $M=(m_{ij})$ to choice of $(x_iy_j)^{m_{ij}}$ in $(1+x_iy_j+(x_iy_j)^2+\cdots+(x_iy_j)^{m_{ij}}+\cdots)$ that term when expanding the product. $\blacksquare$

4/6 Hence the Cauchy identity will follow from the existence of a bij.

$\{n\times n\ \mathbb{N}\text{-matrices } M\} \to \{(P,Q) : \text{SSYTs w/ } \mathrm{sh}(P) = \mathrm{sh}(Q)\}$

s.t. $\mathrm{con}(Q) = (\mathrm{row}_1(M), \mathrm{row}_2(M), \dots)$, $\mathrm{con}(P) = (\mathrm{col}_1(M), \mathrm{col}_2(M), \dots)$ when $M \mapsto (P,Q)$. The Robinson-Schensted-Knuth algorithm is this bijection. ↪ RSK for short

Let's first explain how this is a generalization of RS. The idea is that we encode a permutation $\sigma \in S_n$ by its permutation matrix:

$$\sigma = 213 \longrightarrow \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The $n\times n$ $\mathbb{N}$-matrices w/ row/col sums all $=1$ are exactly the permutation matrices, and similarly, the SSYTs w/ content $=(1,1,\dots,1)$ are exactly the standard tableaux.

$\boxed{\text{RSK applied to a perm. matrix will be RS.}}$

In fact, RSK is only a very slight extension of RS, once we have the correct set-up. ...

The first thing we have to clarify is how to insert into a SSYT.
Now we use the rules: to insert $T \leftarrow i$

- if i is not less than any # in 1^{st} row, put it there,
- otherwise, find leftmost entry it is less than, bump that entry j into the next row, and repeat.

e.g. $\begin{matrix}1&2&2\\3&3&\end{matrix} \leftarrow 1 \xrightarrow{\text{bump } 2} \begin{matrix}1&1&2\\3&3&\end{matrix} \leftarrow 2 \xrightarrow{\text{bump } 3} \begin{matrix}1&1&2\\2&3&\\3&&\end{matrix}$

Next, we need to explain what sequence of #'s we are inserting.
Given matrix M, form biarray $\begin{pmatrix}a_1 & a_2 & \cdots & a_k\\ b_1 & b_2 & \cdots & b_k\end{pmatrix}$ that has $m_{i,j}$ copies of $\binom{i}{j}$ and is s.t. • $a_1 \le \cdots \le a_k$
• $b_i \le b_j$ if $i \le j$ and $a_i = a_j$

e.g. $M = \begin{pmatrix}1&0&\boxed{2}\\0&2&0\\1&1&0\end{pmatrix} \rightsquigarrow$ biarray $\begin{pmatrix}1&1&1&2&2&3&3\\1&\underbrace{3&3}_{m_{1,3}=2}&2&2&1&2\end{pmatrix}$

Then we insert the sequence $b_1, b_2, \dots, b_k$ to form P:

$\emptyset \leftarrow 1$, $\boxed{1} \leftarrow 3$, $\boxed{1\,3} \leftarrow 3$, $133 \leftarrow 2$, $\begin{matrix}1&2&3\\3&&\end{matrix} \leftarrow 2$, $\begin{matrix}1&2&2\\3&3&\end{matrix} \leftarrow 1$, $\begin{matrix}1&1&2\\2&3&\\3&&\end{matrix} \leftarrow 2$

$P = \begin{matrix}1&1&2&2\\2&3&&\\3&&&\end{matrix}$

And what about Q? again it records order new boxes were added to P, but now the entries we add to Q are $a_1, a_2, \dots, a_k$:

$\emptyset$, $\boxed{1}$, $\boxed{1\,1}$, $\boxed{1\,1\,1}$, $\begin{matrix}1&1&1\\2&&\end{matrix}$, $\begin{matrix}1&1&1\\2&2&\end{matrix}$, $\begin{matrix}1&1&1\\2&2&\\3&&\end{matrix}$, $\begin{matrix}1&1&1&3\\2&2&&\\3&&&\end{matrix} = Q$

The map $M \xmapsto{RSK} (P, Q)$ is the RSK algorithm, and it is a straightforward ext. of our arguments about RS to show that it has the desired properties (e.g., is a bijection). ☒

Another construction of RSK via toggles

We will now give a very different description of RSK, which will reveal some hidden symmetries of the algorithm. This does not appear in Sagan. Instead you can read
samuelfhopkins.com/docs/rsk.pdf

To start, we want to encode SSYTs in a different way.

DEF'N A Gelfand-Tsetlin pattern of size n is a triangular array

$$\begin{array}{ccccccccc} g_{1,1} & & g_{1,2} & & g_{1,3} & \cdots & & & g_{1,n} \\ & g_{2,2} & & g_{2,3} & & \cdots & & g_{2,n} & \\ & & g_{3,3} & & \cdots & & g_{3,n} & & \\ & & & \ddots & & \iddots & & & \\ & & & & g_{n,n} & & & & \end{array}$$

of nonnegative integers $g_{i,j} \in \mathbb{N}$ for $1 \leq i \leq j \leq n$

such that $g_{i,j} \geq g_{i+1,j+1} \geq g_{i,j+1}$ $\forall i, j$.

There is a bijection

$$\{\text{SSYTs with entries in } \{1,2,\cdots n\}\} \longrightarrow \{\text{GT patterns of size } n\}$$
$$T \longmapsto GT(T) = (g_{i,j})$$

where $(g_{i,i}, g_{i,i+1}, \cdots g_{i,n}) = sh(T \text{ restricted to entries } \{1,2,\ldots,n+1-i\})$.

e.g. $P = \begin{array}{cccc} 1 & 1 & 2 & 2 \\ 2 & 3 & & \\ 3 & & & \end{array}$ SSYT w/ entries $\subseteq [3]$ $\longrightarrow GT(P) = \begin{array}{ccccc} 4 & & 2 & & 1 \\ & 4 & & 1 & \\ & & 2 & & \end{array}$

since $sh\left(\begin{array}{cccc} 1 & 1 & 2 & 2 \\ 2 & 3 & & \\ 3 & & & \end{array}\right) = (4,2,1)$, $sh\left(\begin{array}{cccc} 1 & 1 & 2 & 2 \\ 2 & & & \end{array}\right) = (4,1)$, $sh(1\,1) = (2)$

Exercise: Prove this really is a bijection.

Recall RSK is a bijection

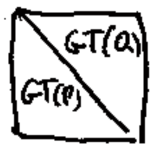
$$\underset{n\times n\ \mathbb{N}\text{-matrix}}{M,} \xmapsto{RSK} (P,Q) \text{ pair of SSYT w/ } sh(P) = sh(Q) \text{ and entries } \subseteq [n]$$

e.g. $M = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix} \xmapsto{RSK} (P,Q) = \left(\begin{matrix} 1 & 1 & 2 & 2 \\ 2 & 3 & & \\ 3 & & & \end{matrix} \;,\; \begin{matrix} 1 & 1 & 1 & 3 \\ 2 & 2 & & \\ 3 & & & \end{matrix} \right)$ which have

$GT(P) = \begin{matrix} 4 & & 2 & & 1 \\ & 4 & & 1 & \\ & & 2 & & \end{matrix} \qquad GT(Q) = \begin{matrix} 4 & & 2 & & 1 \\ & 3 & & 2 & \\ & & 3 & & \end{matrix}$

Notice that since $sh(P) = sh(Q)$, 1^{st} rows of $GT(P)$, $GT(Q)$ are same.

So we can glue $GT(P)$ and $GT(Q)$ into a matrix:

e.g. 

$= \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 2 & 4 & 4 \end{pmatrix}$ ← observe: inequalities on GT-pattern become: weakly increasing along rows + cols!

In other words, we can view RSK as a bijection:

$\{ n \times n\ \mathbb{N}\text{-matrices } M \} \xrightarrow{RSK} \{$ reverse plane partitions π of shape $n \times n \}$.

But what properties does this bij. satisfy?

Recall that $con(P) = col(M)$, and notice that

Sum of i^{th} row (from bottom) of $GT(P)$ = # entries in $\{1, 2, \dots, i\}$ in P

sum of i^{th} lower diagonal of π $=$ sum of 1^{st} i columns of M.

e.g. $\pi = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 2 & 4 & 4 \end{pmatrix}$ $\qquad M = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix}$

2^{nd} diag. $\longrightarrow 4 + 1 = 1 + 2 + 1 + 1$ in two columns in M

Similarly, the i^{th} upper diagonal sum of π $=$ sum of 1^{st} i rows of M.

e.g. $\pi = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 2 & 4 & 4 \end{pmatrix}$ $\qquad M = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix}$ one row in M.

1^{st} diag $\longrightarrow 3 = 1 + 2$

4/11

We will give another construction of this map $M \xmapsto{RSK} \pi$ which converts row/col sums to diagonal sums.

Actually, we will define an even more general bijection.

<u>Thm</u> For any partition shape λ $\exists$ <u>bijection</u>

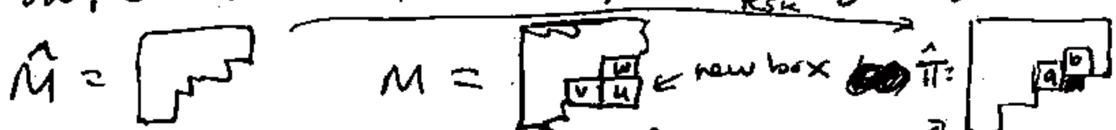
$\{\mathbb{N}\text{-fillings of } \lambda\ M\} \xrightarrow{RSK} \{\text{rev. plane partitions } \pi\ sh^{\text{inner}} = \lambda\}$ — SE boundary

s.t. $\forall$ boxes u on <u>SE</u> <u>ribbon</u> <u>boundary</u>:

- <u>rectangle</u> <u>sum</u> of M at u = <u>diagonal</u> <u>sum</u> of π at u:

<u>e.g.</u> $\lambda = (2,1)$ $\begin{array}{|c|c|}\hline a & b \\ \hline c & \\ \cline{1-1}\end{array} \xmapsto{RSK} \begin{array}{|c|c|}\hline a & a+b \\ \hline a+c & \\ \cline{1-1}\end{array}$

We define RSK <u>recursively</u>. Suppose $\hat{M} \xmapsto{RSK} \hat{\pi}$ for $\hat{\lambda}$, shape obtained from λ by removing single box:



Then define π from $\hat{\pi}$ by:

- <u>toggling</u> all boxes in diagonal of the new box
- filling the new box w/ $\max(a,b) + u$

Here <u>toggling</u> an entry of an r.p.p. does:

$$\begin{array}{ccc} & w & \\ u & \boxed{x} & y \\ & z & \end{array} \mapsto \begin{array}{ccc} & w & \\ u & \boxed{x'} & y \\ & z & \end{array} \quad \text{where } \boxed{x' = \max(u,w) + \min(y,z) - x}$$

<u>Exercise</u>: toggling maintains order for r.p.p.

Then we define $M \xmapsto{RSK} \pi$.

e.g.

$\begin{array}{|c|c|}\hline a & b \\ \hline c & \\ \cline{1-1}\end{array} \xmapsto{RSK} \begin{array}{|c|c|}\hline a & a+b \\ \hline a+c & \\ \cline{1-1}\end{array}$ so $\begin{array}{|c|c|}\hline a & b \\ \hline c & d \\ \hline\end{array} \xmapsto{RSK} \begin{array}{|c|c|}\hline \min(b,c) & a+b \\ \hline a+c & d+\max(b,c)+a \\ \hline\end{array}$

<u>Exercise</u>: Check $M = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix} \xmapsto{RSK} \pi = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 2 & 4 & 4 \end{pmatrix}$ using toggles!

To show that the toggle definition of RSK:
- doesn't depend on order we remove boxes,
- is a bijection,
- converts rectangle sums to diagonal sums

is relatively easy via induction. See my write-up.
To show that $\boxed{\text{toggle RSK} = \text{insertion RSK}}$ is quite involved! But it is true...

And toggle RSK makes one symmetry clear:

<u>Thm</u> If $M \xmapsto{RSK} (P, Q)$ then $M^t \xmapsto{RSK} (Q, P)$.
transpose

<u>Pf</u>: At level of r.p.p.'s, says that $M^t \xmapsto{RSK} \pi^t$, and this is obvious from toggle description! ⊠

<u>Hint</u>: this might be useful on a HW problem...

One final observation is that if $M = (m_{ij}) \xmapsto{RSK} \pi$ then $\sum_{(i,j) = u \in \lambda} h(u) \cdot m_{i,j} = |\pi| = \sum \pi_{i,j}$.

<u>Exercise</u>: Prove from the properties about rectangle and diagonal sums.

So... this "toggle RSK" gives another pf. of:

$$\sum_{\pi \in RPP(\lambda)} q^{|\pi|} = \prod_{u \in \lambda} \frac{1}{1 - q^{h(u)}}$$

<u>But</u> it is <u>not</u> the same bijection as Hillman-Grassl!