Part III Dimensions of the V_λ.

Several formulas exist, we'll derive the hook length formula.

Def. Let λ be a partition. The hook length of a box b in D_λ is the # of boxes directly below or directly to the right of b, counting b once.

Example: This diagram has boxes filled with corresponding hook lengths:

5	4	3	1
3	2	1	

Def: A hook function on D_λ is a function

$\{\text{boxes } b \text{ of } D_\lambda\} \to \{\text{integers } i_b\}$ such that
$-(\#\text{ of boxes directly below } b) \leq i_b \leq (\#\text{ of boxes directly to the right})$

Let $H_\lambda = \{\text{hook functions on } D_\lambda\}$
Then $|H_\lambda| = \prod_{b \in D_\lambda} (\text{hook length of } b)$.

For example, $|H_{4,3}| = 5 \cdot 4 \cdot 3 \cdot 3 \cdot 2 = 360$

Claim: $|\Theta_\lambda| = |\Theta^s_\lambda|\,|H_\lambda|$

This Implies that $\dim(V_\lambda) = n! / \prod(\text{hook lengths})$

(This argument comes from Novelli, Pak, Stoyanovskii)

Let h_0 be the zero valued hook function.

We will construct a bijection

$$\Theta_\lambda \times \{h_0\} \longleftrightarrow \Theta_\lambda^s \times H_\lambda.$$

This establishes the claim.

Let λ be a partition of n.

Let $I = \{$ordered pairs of indices for boxes in $D_\lambda\}$

~~Let s(i) denote the su~~

Canonically order the $i \in I$ in <u>column-decreasing</u> order, e.g.

6	3	1
5	2	
4		

Let $s(i)$ denote the successor to $i \in I$.

The order of business shall be as follows:

1) for each tableau T, construct a standard tableau T_s inductively. That is, given a tableau with no row or column inversions at or before $i \in I$, produce a tableau with no row or column inversions at or before $s(i)$
2) Keep track of extra data to get an injective function (This data is encoded in a hook function).
3) Show that the function is surjective.

<u>Inductive Standardization</u> (no relation to P_λ, Q_λ)

Suppose there are no row or column inversions for $i \in I$ with $i \leq t$. ~~Then let $s(t) = (i,$~~ Set $(i,j) = s(t)$.

Repeat the following procedure until T does not have an inversion at (i,j):

If $b_{i+1,j} < b_{i,j}$ or $b_{i,j+1} < b_{i,j}$, let (i',j')

denote the indices of the smaller of the two values. swap $b_{i,j}$ and $b_{i',j'}$, and then set (i,j) to (i',j').

~~When~~ For example:

$$\begin{array}{|c|c|c|}\hline \boxed{5} & 1 & 4 \\ \hline 2 & 3 & \\ \cline{1-2}\end{array} \rightarrow \begin{array}{|c|c|c|}\hline 1 & \boxed{5} & 4 \\ \hline 2 & 3 & \\ \cline{1-2}\end{array} \rightarrow \begin{array}{|c|c|c|}\hline 1 & 3 & 4 \\ \hline 2 & \boxed{5} & \\ \cline{1-2}\end{array}$$

Notes:

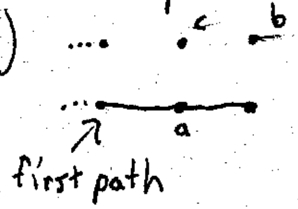
1) at the end of the procedure, $(i,j) = \boxed{5}$ cannot be a column inversion.
2) each time we swap (i,j) with an elt to the right or below, we make the unique choice that ~~does~~ avoids making some element less than $\boxed{5}$ a column inversion.
3) at the ~~very~~ end, no inversions up to $s(t)$, as desired.
4) Running the steps in reverse, $\boxed{5}$ always moves up or left, toward the <u>larger</u> of the two possibilities.
5) We can reverse a step if we know <u>what square we just moved</u> and the ~~what the start~~ entire sequence if we also know the starting position (i,j) for the sequence.
6) The exact value on the box $\boxed{}$ affects when we stop, but not what path we take.

7) We can perform this process $n-1$ times to get a standard tableau.

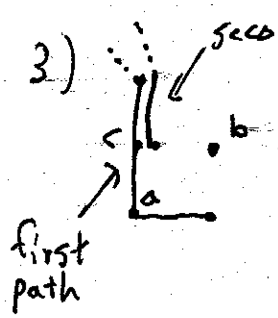
Making the Process injective

Notice something interesting about the paths the elements travel:

1) The path beginning at (i,j) starts above the path beginning at $(i+1,j)$.

2) 

If the second path has been constructed to c in this picture, it must move to b, since prior to the first path being travelled, a was below b, so $b<a$.

3) 

In this picture, $b<a$ again, so the second path must turn right, and go to b.

Def: Given two down-and-rightward paths in D_λ, p_1 is to the right of p_2 if for every $(i_1,j_1)\in p_1$ and $(i_2,j_2)\in p_2$ s.t. $i_1=i_2$, $j_1\geq j_2$.

The above proves this:

Lemma: The path of (i,j) is to the right of the path of $(i+k,j)$ for all $k>0$.

~~For reverse paths the situation is slightly more subtle. but similar. If the reverse paths of (i,j) and (i,j)~~

Thought Experiment:

Assume:

1) we have standardized up to (i,j)
2) we sent elts $(i,j), (i+1,j) \ldots$ in column j, which is of length $i+k-1$, to ~~elts~~ positions $e_1, \ldots, e_k$, but have forgotten which went to which.

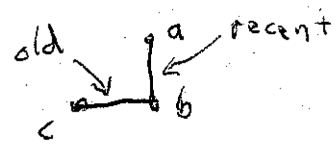
Can we recover the information? (Answer: Almost)

Def: The <u>reverse path</u> $\text{revpath}_{(i,j)}((i',j'))$ of an element at (i',j') with respect to a position (i,j) is the set of positions given by a maximal sequence of upward and leftward moves on squares $\leq (i,j)$, starting at (i',j') and moving toward the box with the larger value when there is a choice.

Notes: 1) A reverse path terminates at (i,j) or $(1,j+1)$.
2) if we standardize at (i,j) The path $(i,j) \to (i',j')$ is the same as $\text{revpath}_{(i,j)}((i',j'))$.

<u>Lemma</u>: The reverse paths of $e_1, \ldots, e_k$ terminate at (i,j)

<u>Pf</u>: For the most recently moved e_i, true by note (2) above. For other e_j, these backtrack along the path originally taken until a more recent path is hit. Then they switch to the newer path

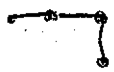


switch at b.

$a > c$, since a <u>was</u> to the right of c.

Define an order on reverse paths terminating at (i,j) as follows:

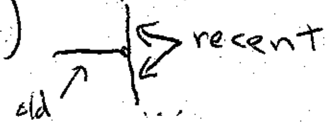
For each reverse path, write a string of d's and r's which describe the "reverse" of the reverse path

e.g.  = rrd

The order is lexicographic with $r >$ blank $> d$.

The ~~greatest~~ path is the most recent, except in the following cases

1) two paths ended at the same position (rev paths are identical)

2) 

In either case, if (i',j') is the "end" of the old reverse path w.r.t. (i,j) then $\text{revpath}_{(i,j)}(i'-1,j'-1) < \text{revpath}_{(i,j)}([\text{recent terminus}])$.

Formally:

* **Lemma:** Suppose (i',j') lies on or below the path of (i,j), then $(i'-1,j'-1)$ lies on or below the path of $(i-1,j)$. If $(i'-1,j'-1)$ lies on the path of $(i-1,j)$, then that path moves to the right at $(i'-1,j'-1)$.

Pf: Follows from the fact that the path at $(i-1,j)$ is to the right of the path at (i,j).

(New Standardization Procedure)

Input: A tableau which has no inversions upto t and a hook function which is zero after t

Output: " " " " " " " " $s(t)$
" " " " " " " " $s(t)$

Let $s(t) = (i, j)$. Define the new tableau T' as before and suppose (i,j) is moved to $(i+\delta_i, j+\delta_j)$. Define a new hook function h' as follows;

$\forall i'$ s.t. $i \le i' < i + \delta_i$ $\quad h'(i', j) := h(i+1, j) - 1$

$h'(i+\delta_i, j) := \delta_j$

$h'(i', j') = h(i', j')$ otherwise.

<u>Def</u>: Let h be a hook function. A <u>special point</u> w.r.t. (i,j) is a position $(i+\delta, j+\varepsilon)$ for $\delta \ge 0$ s.t. $h((i+\delta, j)) = \varepsilon \ge 0$

<u>Prop</u>: Suppose T has been standardized to $t = (i,j)$. The most recently constructed special point has the greatest reverse path.

<u>Pf</u>: Suppose (i,j) is sent to (i',j'). Special points below row i' were special points at the last stage and ~~intersect~~ approach the reverse path of (i',j') from below. Special points above row i' have reverse paths which are smaller by the $*$ lemma.

Corollary: The standardization procedure has a one-sided inverse, and is injective.

Surjectivity:

We need to show that the inverse of the standardization procedure is well-defined starting from an arbitrary hook function and a standard tableau. We still know that there will be a unique greatest reverse path, and that reverse paths don't cross. However, we still need to guarantee that the end of the greatest reverse path at stage (i,j) is (i,j) and not $(1,j+1)$.

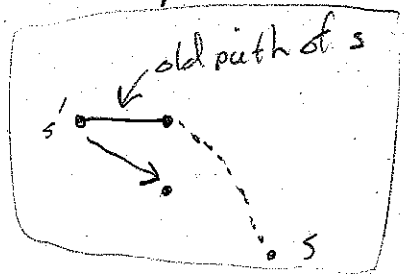
Formally:

Prop: Let T be a Tableau, standard up to $s((i,j))$ and h a hook function which is 0 after $s((i,j))$, such that the greatest special point has a reverse path to $s((i,j))$.
Then, after an inverse standardization step, the new greatest special point has a path to (i,j).

(sketch of) Pf: If $i=1$, clear.
Otherwise, let s be the destandardized special pt from the last step
1) every newly created special pt has $j=0$

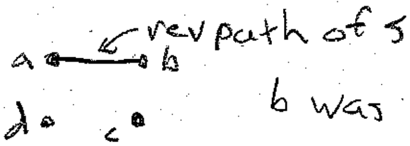
not in a row below s

2) moving a special pt $s' < s$ down and to the right once leaves it on or below the old reverse path of s



3) points s' below s meet the reverse path of s from below.

Now, after moving s to $s((i,j))$ points s' become "stuck below" the reverse path of s.



b was above d, so $b<d$. c is forced to move to d.

Thus, revpath(s') must hit column j below row $i-1$. Thus revpath(s') ends at i.