Fields Schubert.
9 June 2005

Nantel Bergeron, "Schubert polynomials"

Everything for type A:

Recall:

$$\begin{tikzcd} H^*(\mathcal{F}l_{n+1}) \arrow[r, "\cong"] \arrow[d] & R_{n+1} = \dfrac{\mathbb{Z}[x_1, \dots, x_{n+1}]}{\langle e_1, e_2, \dots, e_{n+1} \rangle} \arrow[d] \\ H^*(\mathcal{F}l_n) \arrow[r] & R_n \end{tikzcd}$$

$x_{n+1} \mapsto 0$

$u \leq v$

$u \leq s_i v$

$X_u \subseteq X_{s_i v}$

divided difference op ∂_i

"Go to the limit": $\sigma_u = [X_{u^v}]$ behaves well under these maps — "stable"!

want to find polynomials on RHS, also stable — want to do computations without doing "mod an ideal"

Now just remember RHS: forget the geometry!

$$\partial_i : \mathbb{Z}[x_1, \dots, x_n] \longrightarrow \mathbb{Z}[x_1, \dots, x_n]$$

$$\partial_i(f) = \frac{f - f(\dots x_{i+1}, x_i, \dots)}{x_i - x_{i+1}}$$

Check relations: (easy)

- $\partial_i^2 = 0$
- $\partial_i \partial_j = \partial_j \cdot \partial_i \qquad |i-j| > 1$
- $\partial_i \partial_{i+1} \partial_i = \partial_{i+1} \partial_i \partial_{i+1}$

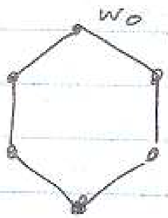
For $w \in S_n$

for reduced $w = s_{a_1} s_{a_2} \cdots s_{a_k}$, can get to other reduced decomp's only using latter 2 relations above.

$\partial_w := \partial_{a_1} \partial_{a_2} \cdots \partial_{a_k}$ well-def'n'd.

A reduced word is a path along Bruhat order diagram

eg: S_3



well-def'n'd means indep of path

Can generate basis of R_n by starting with

$$[x_1^{n-1} x_2^{n-2} \cdots x_n^0]$$

$$\swarrow^{\partial_1} \quad \cdots \quad \searrow^{\partial_2}$$

$$\vdots$$

→ not quite the same as Livin's: he gave van der Monde det, here just take one monomial

<u>Idea</u>: But there should be <u>polynomials</u> representation, not just <u>classes</u>.

Lascoux - Schutzenburger (sp?):

$$\mathfrak{S}_w = \partial_{w_0 w^{-1}} x_1^{n-1} x_2^{n-2} \cdots x_n^0$$

and show nice properties of $\mathfrak{S}_w$, eg stability.

(1) $S_n \hookrightarrow S_{n+1}$

$w \longmapsto w'$ as yesterday

$\mathfrak{S}_w = \mathfrak{S}_{w'}$ <u>stable</u>!

letting $n \to \infty$, $\{\mathfrak{S}_w : w \in S_\infty\}$ basis of $\mathbb{Z}[x_1, x_2, \ldots]$

↑ permute only finitely many.

(2) r last descent of w in S_∞

$\mathfrak{S}_w \in \mathbb{Z}[x_1, \ldots, x_r]$

(3) Expect ± 1's from formula for ∂_i. <u>BUT</u>:

$$\mathfrak{S}_w = \sum_{\alpha = (\alpha_1, \ldots, \alpha_r)} b_{w,\alpha} x^\alpha \qquad (x^\alpha = x_1^{\alpha_1} \cdots x_r^{\alpha_r})$$

<u>MAGIC</u>: $\in \mathbb{Z}_{\geq 0}$!! (some geom. reasons ... ? ask Nantel)

(4) Monk's formula (Type A Chevalley)

$$\mathfrak{S}_{s_k}\mathfrak{S}_w = \sum_{\substack{r \le k < s \\ \ell(w(rs)) = \ell(w)+1}} \mathfrak{S}_{w(rs)}$$

(rs): non-simple transposition

So in this case, coefficients always $= 1$!

Aside: Liviu: $\mathfrak{S}_w = \underbrace{P(\mathfrak{S}_{s_1}, \mathfrak{S}_{s_2}, \dots, \mathfrak{S}_{s_r})}$

THIS is a polynomial, NOT necessarily positive coeff's!! so using this NOT "combinatorial algorithm", doesn't explain positivity.

$$\boxed{\mathfrak{S}_u \mathfrak{S}_v = \sum_w c_{u,v}^w \mathfrak{S}_w}$$

HOLY GRAIL

$c_{u,v}^w \in \mathbb{Z}_{\geq 0}$, by geometry – but want efficient combinatorial algorithms for computing.

Interlude: case of Grassmannian:

v is Grassmannian (at most one descent)

$v = [v(1), v(2), \dots, \quad]$

v has descent at r if $v(r) > v(r+1)$.

if v Grassmannian, can be encoded by following:

perm mx: λ partition

descent r

$$\begin{array}{c} v(1) \\ v(2) \\ \vdots \\ v(r) \\ v(r+1) \\ \vdots \\ \\ \\ \end{array}
\begin{bmatrix}
* & * & 1 & & & & & & \\
* & * & 0 & * & 1 & & & & \\
 & & \vdots & & & & & & \\
* & * & 0 & * & 0 & * & 1 & & \\
1 & 0 & 0 & 0 & 0 & - & - & - & - \\
0 & 1 & 0 & - & - & & & & \\
 & & & 1 & & & & & \\
 & & & & & & 1 & & \\
 & & & & & & & & \ddots
\end{bmatrix}
\begin{array}{c} \\ \\ \\ \\ \\ \left.\begin{array}{c} \\ \\ \\ \end{array}\right\} \text{all 0's} \end{array}$$

Now fill in, 0's to (R) and below..

$= v[\lambda, r]$

$\sigma_{v[\lambda,r]} = s_\lambda(x_1,\dots,x_r)$ Schur function

Recall: always have $Fl \longrightarrow Gr$, $H^*(Gr) \hookrightarrow H^*(Fl)$,
Problem of multiplying these classes completely solved

$$s_\lambda(x_1,\dots,x_r)\, s_\mu(x_1,\dots,x_r) = \sum c_{\lambda\mu}^{\nu}\, s_\nu(x_1,\dots,x_r)$$

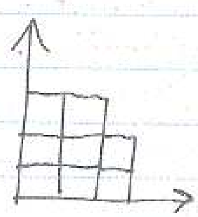
in representation theory ... w/ ∞ variables, then restrict to r. ... so forget some info.
So not quite true that formulas 'same' – Rep thy version has more terms. When $c_{\lambda\mu}^{\nu} = 0$ or 1 different answer for rep thy.

Combinatorics of Schur functions:

$$s_\lambda = \sum_{\substack{T:\lambda\to\{1,\dots,r\} \\ \text{tableaux} \\ \text{(semistandard)}}} \underbrace{\text{monomial}(T)}_{\prod_{s\in\lambda} x_{T(s)} = x^{\mathrm{wt}(T)}}$$

$\lambda = (3,2,2)$

$\lambda \subseteq \mathbb{N}\times\mathbb{N}$



Def^{n}: T semi-std $\iff T(i,j) \le T(i+1,j)$
$T(i,j) < T(i,j+1)$

want then combinatorial approach for

$$s_\lambda s_\mu = \sum c_{\lambda\mu}^{\nu} s_\nu .$$

NB: There are other combinatorial models for the s_λ: Littelmann paths, puzzles etc. ... get different models for symmetries of rules, etc. ($\exists$ 6 symmetries, no model (known) captures all 6.. ?)

"GOALS" : in general case

(A) Find objects R_w

$$\mathfrak{S}_w = \sum_{D \in R_w} x^{wt(D)}$$

∃ many candidates, none of which seem to get at (B) ...

(B) Use it to show 

History

- Kohnert (Bad & Bad)

Given w, draw matrix as yesterday: perm mx, 0's to (R) and below, bubbles for the rest.

bubbles get a "diagram"

$$x^D = \prod x_i^{\#\text{ bubble in row } i}$$

$$\mathfrak{S}_w = \sum_{\substack{\text{bubble} \\ B}} x^B$$

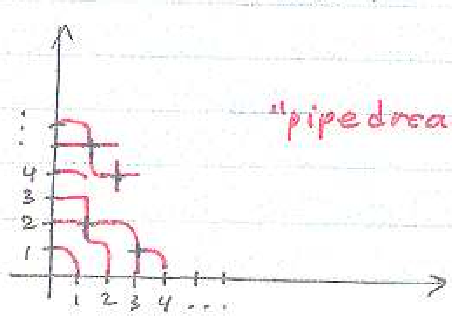
Bad #1 : looking at bubble B, can't tell where it came from, don't know where going

Bad #2 : hard to prove!

- Nautel (also Bad & Bad – but proven)
- Nautel - Sara Billey (NOT due to Fomin-Kirillov) ⟶ some history to set straight here...

RC-graphs

$D \subseteq \mathbb{N} \times \mathbb{N}$



"pipedream"

Such a D is <u>reduced</u> if no strands cross twice!

Also an obvious permutation! Read off the strands.

$$w(D) = [1, 4, 2, 3, 6, 5, 7, \dots]$$

<u>Thm</u> (B-B): $\mathfrak{S}_w = \sum_{\substack{D \text{ Reduce} \\ w(D)=w}} \prod_{(i,j)\in D} x_i$

This is "Good, Good, OK"...

can recover σ, etc

can prove

can recover Monk's formula and some more ...

... but then crashes & burns!

need more ideas.

<u>Alex</u>: $\exists$ saturation conj. for the $c_{\lambda\mu}^{\nu}$?

Nantel: need polytope interpretation — but a good Q!