



GENERACIÓN DE POLINOMIOS DE SCHUBERT CON COCOA I: DIAGRAMAS DE ROTHE¹

Fernando Novoa

*Departamento de Matemáticas, Pontificia Universidad Javeriana
e-mail: fernando.novoa@javeriana.edu.co*

RESUMEN

Se presenta una implementación en el sistema computacional CoCoA de un programa que realiza el cómputo de los polinomios de Schubert, haciendo uso de los diagramas de Rothe.

Palabras clave: Polinomio de Schubert, diagrama de Rothe, permutación.

Abstract

We present a program for computing Schubert polynomials using Rothe diagrams. This program runs in the computational algebraic system CoCoA,

Key words: Schubert polynomial, Rothe diagram, permutation.

INTRODUCCIÓN

Dos propiedades importantes de los polinomios de Schubert son: primero, que ellos representan fielmente el anillo de clases de cohomología de las variedades bandera y segundo, los polinomios de Schur son un caso particular de los polinomios de Schubert. Esto hace de los polinomios de Schubert objetos matemáticos dignos de ser estudiados. Recientemente, se ha logrado su generalización para anillos de cohomología cuántica, y su expansión en términos de monomios elementales cuánticos. De esta forma se obtienen relaciones sorprendentes entre combinatoria, topología, geometría algebraica, métodos numéricos y física.

Los polinomios de Schubert fueron definidos inicialmente por Lascoux y Schützenberger. El programa que se presenta calcula los diagramas de Rothe asociados a los elementos de S_n , los movimientos permitidos en ellos y hace uso de estos diagramas para el cómputo de los polinomios de Schubert. La presentación moderna de los diagramas de Rothe y su uso en la construcción combinatoria de los polinomios de Schubert se debe a Winkel (Winkel, 95) y los movimientos permitidos sobre los diagramas es una idea de Kohnert. Otros métodos combinatorios se han dado para generar estos poli-

¹ Msc 14N15

nomios, por ejemplo, con grafos rc (Berge-ron and Billey, 1993).

Inicialmente se presentan algunas defini-ciones básicas necesarias para la presenta-ción del programa.

DEFINICIONES Y PROPIEDADES

Sean S_n el grupo simétrico sobre n elemen-tos y $Z[x_1, \dots, x_n]$ el anillo de polinomios en las variables x_1, \dots, x_n con coeficientes en los enteros. Se define el *operador de dife-rencias dividido* ∂_k como

$$\partial_k f = \frac{f - s_k f}{x_k - x_{k+1}} \quad k = 1, \dots, n-1,$$

en donde $f \in Z[x_1, \dots, x_n]$ y

$s_k f(\dots, x_k, x_{k+1}, \dots) = f(\dots, x_{k+1}, x_k, \dots)$ representa la acción natural de S_n sobre $Z[x_1, \dots, x_n]$.

El operador de diferencias dividido satis-face las siguientes propiedades:

1. $\partial_k f$ es simétrico en x_k y x_{k+1}
2. $\partial_k^2 = 0$
3. ∂_k conmuta con funciones simétricas en x_k y x_{k+1} . Es decir si g es simétrica en x_k y x_{k+1} entonces

$$\partial_k (fg) = g \partial_k f.$$

4. En general se tiene

$$\partial_k (fg) = g(\partial_k f) + s_k f(\partial_k g).$$

Además se sabe (Kohnert, 1991) que estos operadores satisfacen las relaciones de Coxeter:

1. $\partial_i \partial_{i+1} \partial_i = \partial_{i+1} \partial_i \partial_{i+1}$
2. $\partial_i \partial_j = \partial_j \partial_i$ si $|i - j| > 1$.

Ejemplo 1. Calculemos $\partial_k(x_k^{s+1})$.

$$\partial_k(x_k^{s+1}) = \frac{x_k^{s+1} - x_{k+1}^{s+1}}{x_k - x_{k+1}} =$$

$$\frac{x_k - x_{k+1}}{x_k - x_{k+1}} \left(\sum_{j=0}^s x_k^{s-j} x_{k+1}^j \right) = \sum_{j=0}^s x_k^{s-j} x_{k+1}^j$$

Así, en el caso que $k = 4$ y $s = 3$ tenemos

$$\partial_4(x^4) = \sum x_4^{3-j} x_5^j = x_4^3 + x_4^2 x_5 + x_4 x_5^2 + x_5^3$$

Ejemplo 2. Con el uso de algunas de las rutinas del programa **schubert.pkg** po-demos calcular el operador de diferen-cias dividido. En la primera instrucción se define el anillo de polinomios $Q[x_1, \dots, x_6]$ para calcular $\partial_4(x_4^4)$.

Luego se verifica que en efecto $\partial_k^2 = 0$.

Use `Q[x[1..6]]`;

```
DiDiOperator(4,6,x[4]^4);
x[4]^3 + x[4]^2x[5] + x[4]x[5]^2 + x[5]^3
```

```
DiDiOperator(4,6,DiDiOperator(4,6,x[4]^4));
0
```

Note además que si se intercambian x_k y x_{k+1} el resultado sólo cambia de signo y por lo tanto se tiene que $\partial_k(s_k f) = -\partial_k f$.

Dada $w \in S_n$, es usual escribir la permuta-ción en notación de una línea: $w = [w_1, \dots, w_n]$ en donde $w(i) = w_i$ para todo $i = 1, \dots, n$. Se define la *longitud* de w como el número de inversiones de w , es decir,

$$l(w) = \#\{(i, j) : j > i, w_i > w_j\}.$$

Una descomposición reducida de w es una secuencia (a_1, \dots, a_p) con $1 \leq a_i < n$ tal que

$$w = s_{a_1} \dots s_{a_p}$$

donde, $s_i = (i, i+1)$ y $l(w) = p$. Adicional-mente, denotamos por w_0 a la permutación



de longitud máxima $w_0 = [n, n-1, \dots, 1]$ cuya longitud es $\binom{n}{2}$.

Ejemplo 3. Sea $w = [2, 4, 3, 1]$. Su longitud es $l(w) = 4$. Note que

$$w = (1,2)(2,3)(3,4)(2,3) = s_1 s_2 s_3 s_2.$$

La multiplicación de los ciclos se realiza de derecha a izquierda y claramente para una permutación se pueden tener varias palabras reducidas, debido a las relaciones de Coxeter que también se satisfacen para las permutaciones s_i . Sea $R(w)$ el conjunto de palabras reducidas de w . Supongamos que $(\partial_1, \dots, \partial_p) \in R(w)$. Se define el operador de diferencias ∂_w como

$$\partial_w = \partial_{a_1} \partial_{a_2} \dots \partial_{a_p}.$$

Se sabe que ∂_w es independiente de la descomposición reducida seleccionada (por cuanto los operadores satisfacen las relaciones de Coxeter), por lo tanto el operador está bien definido.

Se define el *polinomio de Schubert* para cualquier $w \in S_n$ como

$$G_w = \partial_{w^{-1}w_0} (x_1^{n-1} x_2^{n-2} \dots x_{n-1})$$

Ejemplo 4. Tomemos nuevamente $w = [2, 4, 3, 1]$. Sabemos que $w_0 = [4, 3, 2, 1]$ y $w^{-1} = [4, 1, 3, 2]$ por lo tanto $w^{-1}w_0 = [2, 3, 1, 4] = (1,2)(23) = s_1 s_2$, luego

$$G_w = \partial_1 \partial_2 (x_1^3 x_2^2 x_3)$$

Para hacer este cómputo se usa directamente la rutina DiDiOperator1(V,N,f). Los parámetros de esta función son V, un vector o lista cuyas entradas son enteros menores o iguales a N y él se calcula el operador de diferencias dividido sobre el polinomio f.

DiDiOperator1([1,2],4,x[1]^3*x[2]^2*x[3]; x[1]^2*x[2]*x[3] + x[1]*x[2]^2*x[3])

con lo que podemos concluir que

$$G_w = \partial_1 \partial_2 (x_1^3 x_2^2 x_3) = x_1^2 x_2 x_3 + x_1 x_2^2 x_3.$$

Diagramas de Rothe.

Como se ha visto, con el operador de diferencias dividida podemos calcular directamente los polinomios de Schubert. Sólo se necesita hallar una palabra reducida para la permutación $w^{-1}w_0$ y aplicar el operador de diferencias. Winkel (Winkel, 1995) y Garsia (Garsia, 2001) presentan algunos algoritmos para factorizar permutaciones.

El programa que se presenta a continuación tiene una aproximación combinatoria y se basa en la construcción de diagramas de Rothe y de los movimientos permitidos en esos diagramas. A cada uno de estos diagramas se le asocia un monomio, el cual va a contribuir al polinomio de Schubert.

Inicialmente cada $w \in S_n$ se identifica con su *matriz de permutación* en la forma usual, es decir, si $w = [w_1, \dots, w_n]$ entonces, la matriz asociada a w tiene las entradas $(w_j, j) = 1$ y las restantes entradas son cero.

Ejemplo 5. La matriz asociada a la permutación $w = [2, 4, 3, 1]$ es

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Por un *diagrama* se entiende un arreglo de $n \times n$ cajas o simplemente una matriz de tamaño $n \times n$. El *diagrama de Rothe* $B(w)$ asociado a $w \in S_n$ es un diagrama de tamaño $n \times n$, al cual se le han cancelado (eliminado) las entradas correspondientes a los *ganchos* de las entradas no nulas de la matriz asociada a $w \in S_n$. Además se sabe que el gancho de una entrada son las cajas que se encuentran a la derecha de esa caja por

la misma fila, junto a las cajas debajo de la entrada y en la misma columna. La caja correspondiente a esa entrada también se incluye en el gancho. En general, se define el gancho de la entrada (w_j, j) como el conjunto de todas las entradas

$$\{(w_j, j') : j' \geq j\} \cup \{(i', j) : i' \geq w_j\}$$

Ejemplo 6. El diagrama de Rothe para la permutación $w = [2,4,3,1]$ se construye de la siguiente manera. Inicialmente se observa el gancho de una entrada $(1,2)$ de la matriz asociada a w .

*	*	*	*
*			
*			

En forma similar se obtiene el diagrama de Rothe para la permutación dada:

			*
*	*	*	*
*		*	*
*	*	*	*

Y reemplazando cajas canceladas por 0 y cajas no cancelada por 1, el diagrama de Rothe $B(w)$ para la permutación se puede dar en forma matricial como:

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

A cada diagrama B se le asigna un monomio de *evaluación* en las variables x_1, \dots, x_n de la siguiente manera:

$$x^B = x_1^{\beta_1} x_2^{\beta_2} \dots x_n^{\beta_n}$$

Donde β_j es el número de entradas no canceladas en la columna j .

Ejemplo 7. El polinomio de evaluación para el diagrama de Rothe asociado a la permutación $w = [2,4,3,1]$ es

$$x^{B(w)} = x_1 x_2^2 x_3$$

A continuación se definen los movimientos permitidos en un diagrama. Usualmente se distinguen estos movimientos diciendo que son K -movimientos, por cuanto Kohnert fue quien los definió.

Sea (i, j) una entrada de un diagrama de Rothe B (es decir, $B_{(i,j)} = 1$), tal que

$$\{(i', j) : i' > i\} \cap B = \emptyset.$$

(todas las cajas debajo de (i, j) por la columna j han sido canceladas) y si además el conjunto

$$M_B(i, j) = \{(i, j') : j' < j, (i, j') \in B\}$$

es distinto de vacío, (luego existen cajas canceladas a la izquierda de la entrada (i, j)) por la fila i , entonces la caja (i, j) se puede mover a la posición en $M_B(i, j)$ con mayor número de columna j' , o sea, a la caja cancelada más cercana a la izquierda de (i, j) por la misma fila i .

Ejemplo 8. En el ejemplo 6 se observa el diagrama asociado a $B = B([2,4,3,1])$. La entrada $(3,2)$ no está cancelada y que las cajas debajo de ella por la misma columna lo están, mientras que a la izquierda de ella por la misma fila hay cajas canceladas. Por lo tanto podemos mover dicha caja a la izquierda. Así podemos mover la caja $(3,2)$ a la posición $(3,1)$ que es la más cercana caja cancelada. El diagrama de este movimiento será

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

y el monomio de evaluación del nuevo diagrama es $x_1^2 x_2 x_3$.

Sea $K(w)$ el conjunto de todos los diagramas de cajas obtenidos del diagrama de Rothe de w , junto con los de sus movimientos y los de los movimientos generados por los movimientos del diagrama y así sucesivamente. Entonces,

Teorema: (Winkel, 1995) El polinomio de Schubert para la permutación $w \in S_n$ está dado por

$$G_w = \sum_{B \in K(w)} x^B$$

Ejemplo 9. El polinomio de Schubert asociado a la permutación $w = [2,4,3,1]$ ya se ha calculado anteriormente. Es fácil convencerse que el conjunto $K(w)$ está dado por los dos diagramas hallados en los ejemplos 6 y 8 y por lo tanto el polinomio de Schubert es la suma de los dos polinomios de evaluación calculados anteriormente. Así

$$G_{[2,4,3,1]} = x_1 x_2^2 x_3 + x_1^2 x_2 x_3.$$

El programa.

El programa **schubert.pkg** implementa el cómputo de los polinomios de Schubert haciendo uso de diagramas de Rothe. Dada una permutación $w \in S_n$, se calculan el diagrama de Rothe para w , lo mismo que el conjunto de los movimientos permitidos $K(w)$. Teniendo este último conjunto, calculamos los polinomios de evaluación para los diagramas en $K(w)$ y se obtiene el polinomio de Schubert, según el teorema anterior.

Las siguientes funciones hacen parte del programa **schubert.pkg**:

1. **Scb.RotheDiagram(w,n):** calcula el diagrama de Rothe para $w \in S_n$.
2. **Scb.Movimientos(w,n):** calcula el conjunto $K(w)$ a partir del diagrama de Rothe $B(w)$.
3. **Scb.SchubertPolynomial(w,n)** calcula el polinomio de Schubert para w .

Adicionalmente tenemos la rutina **PolydeMatriz(B)** el cual calcula el polinomio de evaluación de una matriz. Los diagramas de Rothe se representan en el programa por medio de matrices. Las entradas canceladas son representadas por 0's y las entradas del diagrama por 1's.

Ejemplo 10. Sea $w = [3,1,5,4,2]$. Su diagrama de Rothe es

Scb.RotheDiagram([3,1,5,4,2],5);

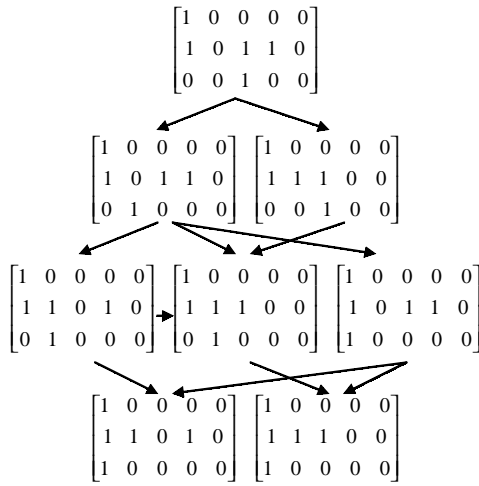
```
Mat[
[1, 0, 0, 0, 0],
[1, 0, 1, 1, 0],
[0, 0, 0, 0, 0],
[0, 0, 1, 0, 0],
[0, 0, 0, 0, 0]
]
```

Ahora se calculan los movimientos permitidos en este diagrama. Estos movimientos los calculamos con el programa **schubert.pkg** y la instrucción **Movimientos** (w,n) cuyos argumentos son $w \in S_n$, y n .

Scb.Movimientos([3,1,5,4,2],5);

```
[
Mat[[1, 0, 0, 0, 0],[1, 0, 1, 1, 0],[0, 0, 0, 0, 0],[0, 0, 1, 0, 0],[0, 0, 0, 0, 0]],
Mat[[1, 0, 0, 0, 0],[1, 0, 1, 1, 0],[0, 0, 0, 0, 0],[0, 1, 0, 0, 0],[0, 0, 0, 0, 0]],
Mat[[1, 0, 0, 0, 0],[1, 1, 1, 0, 0],[0, 0, 0, 0, 0],[0, 0, 1, 0, 0],[0, 0, 0, 0, 0]],
Mat[[1, 0, 0, 0, 0],[1, 0, 1, 1, 0],[0, 0, 0, 0, 0],[1, 0, 0, 0, 0],[0, 0, 0, 0, 0]],
Mat[[1, 0, 0, 0, 0],[1, 1, 0, 1, 0],[0, 0, 0, 0, 0],[0, 1, 0, 0, 0],[0, 0, 0, 0, 0]],
Mat[[1, 0, 0, 0, 0],[1, 1, 0, 1, 0],[0, 0, 0, 0, 0],[0, 1, 0, 0, 0],[0, 0, 0, 0, 0]],
Mat[[1, 0, 0, 0, 0],[1, 1, 0, 1, 0],[0, 0, 0, 0, 0],[1, 0, 0, 0, 0],[0, 0, 0, 0, 0]],
Mat[[1, 0, 0, 0, 0],[1, 1, 1, 0, 0],[0, 0, 0, 0, 0],[1, 0, 0, 0, 0],[0, 0, 0, 0, 0]]
]
```

Estas matrices corresponden a los siguientes 8 movimientos permitidos. Dado que filas de ceros no afectan ni a los monomios de evaluación ni a los movimientos en el diagrama, estas filas se pueden suprimir del diagrama y en este caso tenemos que el diagrama y sus movimientos se representan así:



En total se tienen 8 movimientos distintos y de cada uno de éstos se obtiene un monomio de evaluación que contribuirá al polinomio de Schubert. Con el programa que presentamos este polinomio se calcula con la instrucción **SchubertPolynomial** (w,n), donde $w \in S_n$.

```
Scb.SchubertPolynomial([3,1,5,4,2],5);
x[1]^3x[2]x[3] + x[1]^2x[2]^2x[3] +
x[1]^2x[2]x[3]^2 + x[1]^3x[2]x[4] +
x[1]^2x[2]^2x[4] + x[1]^3x[3]x[4] +
x[1]^2x[2]x[3]x[4] + x[1]^2x[3]^2x[4]
```

CONCLUSIONES

El programa **schubert.pkg** calcula los polinomios de Schubert por medio de la representación de los diagramas de Rothe y está realizado para el sistema algebraico computacional CoCoA. Este programa en combinación con el programa **schur.pkg**

para el cómputo de polinomios de Schur, sirven como herramienta computacional en un curso de álgebra en donde se trabajen anillos de polinomios en varias variables y en especial, cambios de bases para anillos de polinomios con coeficientes enteros. Estas familias de polinomios forman bases para diversas familias de anillos, por lo que estos programas pueden ser de gran utilidad cuando sea necesario representar los miembros de una de estas familias o productos de ellos como combinaciones lineales de los elementos de la otra.

Adicionalmente se tienen otros programas para calcular los polinomios skew de Schur, los polinomios de Schubert con operadores de diferencias divididos y con grafos rc. Estos programas y el uso de algunas rutinas de CoCoA nos proveen herramientas fundamentales para comprobar propiedades combinatorias que relacionan los coeficientes de estos polinomios con tablas de Young de diversas clases.

La necesidad de tener herramientas para determinar los diagramas de Rothe no es simplemente para calcular polinomios de Schubert. Estos diagramas poseen además propiedades combinatorias que ayudan a explicar la construcción y determinación los polinomios de Grothendieck y algunas soluciones para ecuaciones del tipo Yang-Baxter (Lascoux, 2002).

LITERATURA CITADA

BERGERON N. AND BILLEY S. *RC-GRAPHS AND SCHUBERT POLYNOMIALS*. PREPRINT. 1993.

BILLEY S., JOCKUSH W. AND STANLEY R. *SOME COMBINATORIAL PROPERTIES OF SCHUBERT POLYNOMIALS*. PREPRINT, 1993.

FULTON W. *YOUNG TABLEAUX*. LMSST 35 CAMBRIDGE UNIVERSITY PRESS. 1997.

GARSIA A.M. *THE SAGA OF REDUCED FACTORIZATIONS ELEMENTS OF THE SYMMETRIC GROUP*.



Julio-diciembre de 2003

TOPICS IN ALGEBRA, LECTURE NOTES, APRIL 4, 2001.

KOHNERT A. *SCHUBERT POLYNOMIALS AND SKEW SCHUR FUNCTIONS*. PREPRINT. 1991.

LASCOUX A. *CHEMN AND YANG THROUGH ICE*. PREPRINT. 2002.

MACDONAL I.G. *SYMMETRIC FUNCTIONS, AND ORTHOGONAL POLYNOMIALS*. AMERICAN MATHEMATICAL SOCIETY, PROVIDENCE, RI. 1998.

MANIVEL L. *SYMMETRIC FUNCTIONS, SCHUBERT POLYNOMIALS AND DEGENERACY LOCI*. AMERICAN MATHEMATICAL SOCIETY, PROVIDENCE. 2001.

WINKEL R. *ON THE EXPANSION OF SCHUR AND SCHUBERT POLYNOMIALS INTO STANDARD ELEMENTARY MONOMIALS*. PREPRINT. 1997.

WINKEL R. *DIAGRAM RULES FOR GENERATION OF SCHUBERT POLYNOMIALS*. PREPRINT. 1995.

Recibido: 06-02-2003

Aceptado: 16-08-2003

