Ábacos, Particiones Estables y la Conjetura de Alperin para los Grupos Simétricos en Característica Dos

Isa Ingrid Acuña Arroyo

mayo de 2006

Índice general

Introducción

Esta es una tesis de combinatoria y computación, con cierta relación con álgebra, por lo que las definiciones y resultados que se requieren de álgebra se pueden ver al final, en el apéndice algebraico.

En este trabajo empezaremos por definir lo que son las particiones, veremos lo que son ganchos, ganchos sesgados, las longitudes de los ganchos de la primera columna, el 2-corazón de una partición y algunas cosas que se pueden hacer con ellas, luego definiremos el ábaco de James, veremos cómo representar particiones en el ábaco de James y observaremos ahí que es lo mismo quitar ganchos sesgados de una partición, que recorrer todas las cuentas hacia arriba en la representación de la partición en el ábaco de James. En el cuarto capítulo definiremos lo que son las particiones 2-estables, después veremos algunos resultados originales, el más importante es que refutamos la unicidad de la matriz de particiones dada en el artículo Radha Kessar - Luis Valero Elizondo, Stable partitions and Alperin's weight conjecture for the symmetric groups in characteristic two, Boletín Sociedad Matemática Mexicana (3) Vol. 10, (2004), 53-62, donde se conjetura la existencia y unicidad de una matriz cuyas entradas son particiones que tengan las siguientes propiedades:

1. Que las particiones se vayan conteniendo por renglones.
2. Que su 2-corazón esté indexado por la columna.
3. Que sean particiones 2-regulares.
4. Que dentro de un mismo renglón la diferencia de los números de nodos de dos particiones es igual a la diferencia del número de nodos de sus respectivos dos corazones.

En el capítulo 6, veremos el código en Gap que se usó para correr algunos ejemplos y rutinas que pueden formar una biblioteca para trabajar con particiones y ábacos.

En el capítulo 7 veremos las conclusiones y finalmente el apéndice algebraico donde vienen los resultados y definiciones de álgebra que se utilizan para la tesis.

Capítulo 1

Particiones

En este capítulo empezaremos por ver que una partición de un número n no es más que una lista finita de números naturales tales que su suma es igual a n. Este material se puede encontrar en James-Kerber.

Definición: Una sucesión de enteros no negativos $\alpha = \{\alpha_1, \alpha_2, \ldots\}$ es una **partición** de un entero positivo n, denotado $\alpha \vdash n$, si

1. Para toda $i \geq 1, \alpha_i \geq \alpha_{i+1}$;
2. $\sum_{i=1}^{\infty} \alpha_i = n$.

Los α_i se llaman las **partes** de la partición α.

Ya que sabemos lo que es una partición, ahora necesitamos poder expresarla de alguna manera visual, para eso definimos lo que es el diagrama de la partición α, que será el dibujo que usaremos para "ver" las particiones.

Definición: El **diagrama** $[\alpha]$ de la partición α es un dibujo que consta de α_1 **cuadros** o **nodos** en el primer renglón, α_2 cuadros en el segundo renglón y así sucesivamente hasta llegar a α_k nodos en el k-ésimo renglón.

Veamos algunos ejemplos de particiones y sus diagramas correspondientes.

Ejemplos:

1. $(5,3,1) \vdash 9$
$$\begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & \times & & \\ \times & & & & \end{array}$$
2. $(5) \vdash 5$
$$\begin{array}{ccccc} \times & \times & \times & \times & \times \end{array}$$
3. $(3,3) \vdash 6$
$$\begin{array}{ccc} \times & \times & \times \\ \times & \times & \times \end{array}$$
4. $(6,4,2,2) \vdash 14$
$$\begin{array}{cccccc} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & & \\ \times & \times & & & & \\ \times & \times & & & & \end{array}$$

Ya que vimos cómo se ve el diagrama de una partición, ahora definamos qué es el gancho de una partición correspondiente a un nodo (i,j), es decir, al nodo que se encuentra en el renglón i y la columna j del diagrama de la partición.

Definición: Sea $\alpha = \{\alpha_1, \ldots, \alpha_k\}$ una partición de n. El nodo (i,j) es el nodo que se encuentra en el renglón i y la columna j. Definimos el **gancho** de la partición α correspondiente al nodo (i,j) y lo denotamos H_{ij}^{α} como el nodo (i,j), todos los nodos a la derecha del nodo (i,j), esto es, $(i,j), \ldots, (i,\alpha_i)$ y todos los nodos hacia abajo del nodo (i,j) en la columna j, esto es, $(i,j), \ldots, (k,j)$.

Ya que sabemos lo que es el gancho de una partición correspondiente a un nodo (i,j), tiene sentido definir el tamaño del gancho.

Definición: El **tamaño** de un gancho es el número de nodos en el gancho. Un q**-gancho** es un gancho de tamaño q.

Entonces un gancho se compone de dos partes: los nodos que están a la derecha del nodo (i,j) y los nodos que se encuentran abajo del nodo (i,j); a estas partes las definiremos como brazo y pie del gancho respectivamente.

Definición: Definimos **brazo** del gancho de una partición correspondiente al nodo (i,j), como todos los nodos en el renglón i que se encuentran a la derecha del nodo (i,j).

Definición: Definimos **pie** del gancho de una partición correspondiente al nodo (i,j), como todos los nodos en la columna j que se encuentran abajo del nodo (i,j).

Veamos ahora algunos ejemplos de ganchos y cómo se ven los ganchos en los diagramas.

Ejemplos:

1. $(6,4,3,2,1,1) = \alpha \vdash 17$

$$\begin{array}{llllll} \times & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \times & \boxtimes & \times & \times & & \\ \times & \boxtimes & \times & & & \\ \times & \boxtimes & & & & \\ \times & & & & & \\ \times & & & & & \end{array}$$

   $H^{\alpha}_{1,2}$ es un $8 - gancho$

2. $(5,3,1) = \alpha \vdash 9$

$$\begin{array}{lllll} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \times & \times & & \\ \boxtimes & & & & \end{array}$$

   $H^{\alpha}_{1,1}$ es un $7 - gancho$

3. $(4,3,3,1) = \alpha \vdash 11$

$$\begin{array}{llll} \times & \times & \times & \times \\ \times & \boxtimes & \boxtimes & \\ \times & \boxtimes & \times & \\ \times & & & \end{array}$$

   $H^{\alpha}_{2,2}$ es un 3-gancho.

4. $(6,3,1,1) = \alpha \vdash 11$

$$\begin{array}{llllll} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & & & \\ \boxtimes & & & & & \\ \boxtimes & & & & & \end{array}$$

$H^{\alpha}_{3,1}$ es un 2-gancho.

Ya que vimos lo que es el gancho de una partición, veamos ahora lo que es el gancho sesgado de la partición α correspondiente al nodo (i, j).

Definición: Definimos el gancho **sesgado** de la partición α correspondiente al nodo (i, j), denotado por R^{α}_{ij} como el conjunto de nodos que unen el último nodo del renglón i con el último nodo de la columna j, incluyendo éstos últimos y tal que la manera de unirlos es avanzando de un nodo hacia otro de manera horizontal y vertical siguiendo los nodos de la orilla derecha de la partición.

Veamos ahora algunos ejemplos de gancho sesgado, ya que esta definición es más complicada.

Ejemplos:

1. $\alpha = (5, 4, 3, 2, 1)$

$$\begin{array}{lllll} \times & \otimes & \times & \boxtimes & \boxtimes \\ \times & \times & \boxtimes & \boxtimes & \\ \times & \boxtimes & \boxtimes & & \\ \times & \boxtimes & & & \\ \times & & & & \end{array}$$

   $R^{\alpha}_{1,2}$ es el gancho sesgado de α correspondiente al nodo $(1, 2)$.

2. $\alpha = (6, 4, 2)$

$$\begin{array}{llllll} \times & \times & \times & \times & \times & \times \\ \otimes & \boxtimes & \boxtimes & \boxtimes & & \\ \boxtimes & \boxtimes & & & & \end{array}$$

   $R^{\alpha}_{2,1}$ es el gancho sesgado de α correspondiente al nodo $(2, 1)$.

3. $\alpha = (7, 5, 5, 1)$

$$\begin{array}{lllllll} \times & \times & \otimes & \times & \boxtimes & \boxtimes & \boxtimes \\ \times & \times & \times & \times & \boxtimes & & \\ \times & \times & \boxtimes & \boxtimes & \boxtimes & & \\ \times & & & & & & \end{array}$$

   $R^{\alpha}_{1,3}$ es el gancho sesgado de α correspondiente al nodo $(1, 3)$.

Ya que nos queda clara la definición de gancho sesgado, es más natural definir el borde de una partición.

Definición: Definimos el **borde** de la partición α como el gancho sesgado de la partición α correspondiente al nodo $(1,1)$.

De la misma manera que definimos el tamaño de un gancho, definimos ahora el tamaño de un gancho sesgado.

Definición: El **tamaño** de un gancho sesgado es igual al número de nodos en el gancho sesgado. Un q-gancho **sesgado** es un gancho sesgado de tamaño q.

Veamos en unos ejemplos lo que es el borde de una partición, o lo que es lo mismo, el gancho sesgado de la partición α respecto al nodo $(1,1)$.

Ejemplos:

1. $\alpha = (5,4,3,2,1)$

$$\begin{array}{ccccc}
\otimes & \times & \times & \boxtimes & \boxtimes \\
\times & \times & \boxtimes & \boxtimes & \\
\times & \boxtimes & \boxtimes & & \\
\boxtimes & \boxtimes & & & \\
\boxtimes & & & &
\end{array}$$

   $R^{\alpha}_{1,1}$ es el borde de la partición α.
   $R^{\alpha}_{1,1}$ es un 9-gancho sesgado.

2. $\alpha = (8,5,5,3,2)$

$$\begin{array}{cccccccc}
\otimes & \times & \times & \times & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
\times & \times & \times & \times & \boxtimes & & & \\
\times & \times & \boxtimes & \boxtimes & \boxtimes & & & \\
\times & \boxtimes & \boxtimes & & & & & \\
\boxtimes & \boxtimes & & & & & &
\end{array}$$

   $R^{\alpha}_{1,1}$ es el borde de la partición α.
   $R^{\alpha}_{1,1}$ es un 12-gancho sesgado.

Finalmente, definiremos una de las partes clave en este trabajo; los q-corazones, aunque a nosotros nos interesan más los 2-corazones.

Definición: Una partición α o un diagrama $[\alpha]$ que no contiene ningún q-gancho sesgado es llamado q**-corazón**.

Veamos ejemplos de cómo se ven los 2-corazones, los 3-corazones y los 4-corazones.

Ejemplos:

1. Los 2-corazones se ven de la forma:

$$
\begin{array}{l}\times\end{array}
\quad
\begin{array}{ll}\times & \times \\ \times & \end{array}
\quad
\begin{array}{lll}\times & \times & \times \\ \times & \times & \\ \times & & \end{array}
\quad
\begin{array}{llll}\times & \times & \times & \times \\ \times & \times & \times & \\ \times & \times & & \\ \times & & & \end{array}
\quad
\begin{array}{lllll}\times & \times & \times & \times & \times \\ \times & \times & \times & \times & \\ \times & \times & \times & & \\ \times & \times & & & \\ \times & & & & \end{array}
$$

2. Los 3-corazones se ven de la forma:

$$
\begin{array}{l}\times\end{array}
\quad
\begin{array}{ll}\times & \times\end{array}
\quad
\begin{array}{l}\times \\ \times\end{array}
\quad
\begin{array}{lll}\times & \times & \times \\ \times & & \end{array}
\quad
\begin{array}{ll}\times & \times \\ \times & \\ \times & \end{array}
\quad
\begin{array}{lll}\times & \times & \times \\ \times & & \\ \times & & \end{array}
$$

3. Los 4-corazones se ven de la forma:

$$
\begin{array}{l}\times\end{array}
\quad
\begin{array}{ll}\times & \times\end{array}
\quad
\begin{array}{l}\times \\ \times\end{array}
\quad
\begin{array}{l}\times \\ \times \\ \times\end{array}
\quad
\begin{array}{ll}\times & \times \\ \times & \end{array}
\quad
\begin{array}{lll}\times & \times & \times\end{array}
\quad
\begin{array}{llll}\times & \times & \times & \times \\ \times & & & \end{array}
\quad
\begin{array}{lll}\times & \times & \times \\ \times & & \\ \times & & \end{array}
\quad
\begin{array}{ll}\times & \times \\ \times & \\ \times & \\ \times & \end{array}
$$

Ahora vamos a tomar la primera columna de la partición y nos fijaremos en las longitudes de los ganchos de cada elemento de esa columna.

Definición: Denotaremos por h_i^α las longitudes de los ganchos de la primera columna de α.

Veamos algunos ejemplos de las longitudes de ganchos de la primera columna de la partición α.

Ejemplos:

1. $\alpha = (5, 4, 3, 2, 1)$

$$\begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \\ \times & \times & \times & & \\ \times & \times & & & \\ \times & & & & \end{array} \qquad \begin{array}{l} h_1^\alpha = 9 \\ h_2^\alpha = 7 \\ h_3^\alpha = 5 \\ h_4^\alpha = 3 \\ h_5^\alpha = 1 \end{array}$$

2. $\alpha = (5, 3, 3, 1)$

$$\begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & \times & & \\ \times & \times & \times & & \\ \times & & & & \end{array} \qquad \begin{array}{l} h_1^\alpha = 8 \\ h_2^\alpha = 5 \\ h_3^\alpha = 4 \\ h_4^\alpha = 1 \end{array}$$

3. $\alpha = (6, 4, 2)$

$$\begin{array}{cccccc} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & & \\ \times & \times & & & & \end{array} \qquad \begin{array}{l} h_1^\alpha = 8 \\ h_2^\alpha = 5 \\ h_3^\alpha = 2 \end{array}$$

En los ejemplos presentados, H_{ij}^α y R_{ij}^α siempre tienen el mismo número de nodos. Esto no es una coincidencia, es un lema que se demostrará a continuación y que equivale a usar la métrica del taxista, (si tenemos un plano en el que sólo podemos avanzar de manera horizontal o vertical) el tamaño de un gancho es equivalente a medir con la métrica del taxista, la distancia de un extremo a otro del gancho; entonces como la distancia es independiente de la trayectoria, y la única diferencia entre gancho y gancho sesgado es precisamente la trayectoria, podemos concluír que tienen el mismo tamaño.

Lema: Para cualquier $\alpha \vdash n$, para toda i y para toda j, H_{ij}^α y R_{ij}^α tienen el mismo número de nodos.

Demostración

Si las únicas direcciones en las que podemos avanzar son vertical y horizontal (esto es, arriba y derecha o abajo e izquierda) entonces la longitud de H_{ij}^α y R_{ij}^α son iguales ya que la suma de los nodos en sentido horizontal de R_{ij}^α es igual a $\alpha_i - i$ que es igual a la suma de los nodos en sentido horizontal de H_{ij}^α. Lo mismo pasa con los nodos en sentido vertical.

$$
\begin{array}{ccccccc}
\times & \times & \times & \times & \otimes & \boxtimes & \boxtimes \\
\times & \times & \times & \otimes & \boxtimes & & \\
\times & \times & \otimes & \boxtimes & & & \\
\times & \times & \otimes & & & & \\
\times & \otimes & \boxtimes & & & & \\
\otimes & \boxtimes & & & & & \\
\bigotimes & & & & & &
\end{array}
$$

∎

Si a una partición α le quitamos un gancho sesgado, el diagrama que nos queda sigue siendo el diagrama de una partición, esto es, sigue cumpliendo con los requisitos para ser partición.

Lema: $[\alpha] - R_{ij}^{\alpha}$ es el diagrama de una partición para toda $\alpha \vdash n$ y para todo R_{ij}^{α}.

Demostración

Al quitar a un diagrama de una partición un gancho sesgado, estamos quitando una parte del borde de la partición de tal manera que $\alpha_{i+1} \leq \alpha_i$ y así $[\alpha] - R_{ij}^{\alpha}$ sigue siendo el diagrama de una partición.

$$
\begin{array}{ccccccc}
\times & \times & \times & \times & \times & \times & \times \\
\times & \times & \bigotimes & \times & \boxtimes & \boxtimes & \\
\times & \times & \times & \boxtimes & \boxtimes & & \\
\times & \times & \times & \boxtimes & & & \\
\times & \times & \boxtimes & \boxtimes & & & \\
\times & \times & & & & &
\end{array}
\qquad
\begin{array}{ccccccc}
\times & \times & \times & \times & \times & \times & \times \\
\times & \times & \times & \times & & & \\
\times & \times & \times & & & & \\
\times & \times & \times & & & & \\
\times & \times & & & & & \\
\times & \times & & & & &
\end{array}
$$

$\alpha = (7, 6, 5, 4, 4, 2)$ $\qquad$ $[\alpha] - R_{2,3}^{\alpha} = (7, 4, 3, 3, 2, 2)$

∎

Fijémonos de nuevo en las longitudes de gancho de la primera columna de α, vamos a probar que es una sucesión estrictamente decreciente, ya que si no lo fuera, estaríamos hablando de una sucesión de números que no es partición.

Lema: Las longitudes de los ganchos de la primera columna de α forman una sucesión estrictamente decreciente.

Demostración

El pie de h_i^α es mayor que el pie de h_{i+1}^α, entonces si $h_i^\alpha = h_{i+1}^\alpha$, el brazo de $h_{i=1}^\alpha$ tendría que ser mayor que el brazo de h_i^α y esto implica que α no es una partición.

$$\begin{array}{llllll} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & & \\ \times & \times & \times & \times & & \\ \times & \times & & & & \\ \times & & & & & \end{array} \qquad \begin{array}{c} h_1^\alpha = 10 \\ h_2^\alpha = 7 \\ h_3^\alpha = 6 \\ h_4^\alpha = 3 \\ h_5^\alpha = 1 \end{array}$$

$$\alpha = (6,4,4,2,1) \qquad h_1^\alpha > h_2^\alpha > h_3^\alpha > h_4^\alpha > h_5^\alpha$$

▮

De la misma manera que, teniendo una partición α, calculamos la sucesión de longitudes de gancho de la primera columna; si tenemos ahora la sucesión de longitudes de gancho de la primera columna de la partición α, podemos recuperar la partición α. Veamos de qué forma.

Lema: Toda partición está determinada por sus longitudes de ganchos de la primera columna.

Demostración

Sean $\alpha = \{\alpha_1, \ldots, \alpha_k\}$ una partición y $h = \{h_1^\alpha, \ldots, h_k^\alpha\}$. Entonces $\alpha_i = h_i^\alpha - (k-i)$.

▮

De hecho, si tenemos cualquier sucesión escrictamente decreciente de números enteros, no negativos, podemos asociarle una partición.

Observación: Dada una sucesión estrictamente decreciente de enteros no negativos $\beta_1 > \beta_2 > \ldots > \beta_r$ podemos crear una partición α haciendo

$$\alpha_i := \beta_i + i - r, \qquad 1 \leq i \leq r.$$

Veamos algunos ejemplos para que quede más clara la idea.

Ejemplos:

1. $\beta = \{9, 7, 5, 3, 1, 0\} \qquad r = 6$

$$\begin{array}{lcccc} \alpha_1 &=& 9+1-6 &=& 4 \\ \alpha_2 &=& 7+2-6 &=& 3 \\ \alpha_3 &=& 5+3-6 &=& 2 \\ \alpha_4 &=& 3+4-6 &=& 1 \\ \alpha_5 &=& 1+5-6 &=& 0 \\ \alpha_6 &=& 0+6-6 &=& 0 \end{array}$$

   Obtenemos la partición $\alpha = \{4, 3, 2, 1\}$, $\alpha \vdash 10$

2. $\beta = \{10, 9, 3, 2, 1, 0\} \qquad r = 6$

$$\begin{array}{lcccc} \alpha_1 &=& 10+1-6 &=& 5 \\ \alpha_2 &=& 9+2-6 &=& 5 \\ \alpha_3 &=& 3+3-6 &=& 0 \\ \alpha_4 &=& 2+4-6 &=& 0 \\ \alpha_5 &=& 1+5-6 &=& 0 \\ \alpha_6 &=& 0+6-6 &=& 0 \end{array}$$

   Obtenemos la partición $\alpha = \{5, 5\}$, $\alpha \vdash 10$

3. $\beta = \{7, 4, 1, 0\} \qquad r = 4$

$$\begin{array}{lcccc} \alpha_1 &=& 7+1-4 &=& 4 \\ \alpha_2 &=& 4+2-4 &=& 2 \\ \alpha_3 &=& 1+3-4 &=& 0 \\ \alpha_4 &=& 0+4-4 &=& 0 \end{array}$$

   Obtenemos la partición $\alpha = \{4, 2\}$, $\alpha \vdash 6$

4. $\beta = \{15, 14, 13, 10, 8, 6, 4, 3, 2, 1, 0\} \qquad r = 11$

$$\begin{array}{lcccc} \alpha_1 &=& 15+1-11 &=& 5 \\ \alpha_2 &=& 14+2-11 &=& 5 \\ \alpha_3 &=& 13+3-11 &=& 5 \\ \alpha_4 &=& 10+4-11 &=& 3 \\ \alpha_5 &=& 8+5-11 &=& 2 \\ \alpha_6 &=& 6+6-11 &=& 1 \\ \alpha_7 &=& 4+7-11 &=& 0 \end{array}$$

Obtenemos la partición $\alpha = \{5^3, 3, 2, 1\}$, $\alpha \vdash 21$

5. $\beta = \{8, 5, 4, 2, 1\} \qquad r = 5$

$$\begin{array}{lllll} \alpha_1 & = & 8+1-5 & = & 4 \\ \alpha_2 & = & 5+2-5 & = & 2 \\ \alpha_3 & = & 4+3-5 & = & 2 \\ \alpha_4 & = & 2+4-5 & = & 1 \\ \alpha_5 & = & 1+5-5 & = & 1 \end{array}$$

Obtenemos la partición $\alpha = \{4, 2^2, 1^2\}$, $\alpha \vdash 10$

6. $\beta = \{16, 13, 10, 5, 3, 2, 1\} \qquad r = 7$

$$\begin{array}{lllll} \alpha_1 & = & 16+1-7 & = & 10 \\ \alpha_2 & = & 13+2-7 & = & 8 \\ \alpha_3 & = & 10+3-7 & = & 6 \\ \alpha_4 & = & 5+4-7 & = & 2 \\ \alpha_5 & = & 3+5-7 & = & 1 \\ \alpha_6 & = & 2+6-7 & = & 1 \\ \alpha_7 & = & 1+7-7 & = & 1 \end{array}$$

Obtenemos la partición $\alpha = \{10, 8, 6, 2, 1^3\}$, $\alpha \vdash 29$

El siguiente lema formaliza la manera de asociar una partición a una sucesión estrictamente decreciente de enteros no negativos.

Lema: Existe una función

$$\phi : \wp_{<\infty}(\mathbb{N}_0) = \{A | A \subset \mathbb{N}_0, A \qquad \text{finito}\} \longrightarrow \{\text{particiones}\}$$

tal que dado un conjunto $\beta = \{\beta_1 > \beta_2 > \ldots > \beta_r\}$ le asigna la partición $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ dada por $\alpha_i = \beta_i + i - r, 1 \leq i \leq r$. Además, $0 \notin \beta$, α tiene $|\beta| = r$ partes $\iff$ β es la sucesión de longitudes de ganchos de la primera columna de α.

Demostración

ϕ está bien definida, es decir, $\phi(\beta) = \alpha = \{\alpha_1, \ldots, \alpha_r\}$ es una partición, pues

$$\alpha_i = \beta_i + i - r > \beta_{i+1} + i - r = \beta_{i+1} + i + 1 - 1 - r = \alpha_{i+1} - 1 \qquad (1.1)$$

Por lo tanto, $\alpha_i \geq \alpha_{i+1}$.
Note que $0 \notin \beta \iff \beta_r > 0 \iff \alpha_r = \beta_r + r - r = \beta_r > 0 \iff \alpha$ tiene $|\beta| = r$ partes.
Antes vimos que si β es la sucesión de longitudes de ganchos de la primera columna de α, entonces $\beta_r = \alpha_r > 0$. Supongamos ahora que β es tal que $\beta_r > 0$, entonces α tiene $|\beta| = r$ partes. Así que sus longitudes de gancho de la primera columna son $h = \{h_1, \ldots, h_r\}$ y cumplen la fórmula $\alpha_i = h_i + i - r, 1 \leq i \leq r$. Pero por otro lado, tenemos que $\alpha_i = \beta_i + i - r$. Por lo tanto, $h_i = \beta_i$ para toda $1 \leq i \leq r$.

∎

Entonces, a una sucesión β estrictamente decreciente de enteros no negativos, le podemos asociar sólo una partición α, pero a una partición α le podemos asociar varias sucesiones β. Veamos algunos ejemplos.

Ejemplos:

1. La partición ϕ se puede asociar a varias β:

$$\beta = \emptyset, \{0\}, \{1, 0\}, \{2, 1, 0\}, \{3, 2, 1, 0\}, \ldots, \{n, n-1, \ldots, 2, 1, 0\}$$

2. La partición $\{1\}$ se puede asociar a:

$$\beta = \{1\}, \{2, 0\}, \{3, 1, 0\}, \{4, 2, 1, 0\}, \ldots, \{n+2, n, n-1, \ldots, 1, 0\}$$

3. La partición $\alpha = \{3, 2, 2\}$ se puede asociar a:

$$\beta = \{5, 3, 2\}, \{6, 4, 3, 0\}, \ldots, \{n+6, n+4, n+3, n, n-1, \ldots, 1, 0\}$$

Veamos ahora la demostración de que dadas las longitudes de gancho de una partición α y una sucesión β asociada a la partición α, la función ϕ definida anteriormente, asocia a ambas la partición α.

Lema: Sean $h, \beta \subset \mathbb{N}_0$, con $0 \notin h = \{h_1 > h_2 > \ldots > h_t\}$ y

$$\beta = \{n+1+h_1, \ldots, n+1+h_{t-1}, n+1+h_t, n, n-1, \ldots, 2, 1, 0\}$$

donde $n \in \mathbb{N}_0$. Entonces $\phi(h) = \phi(\beta)$ y h es la sucesión de longitudes de gancho de la primera columna de $\phi(\beta)$.

Demostración

Escribamos $\beta = \{\beta_1 > \beta_2 > \ldots > \beta_s\}$, donde $s = t + n + 1$. Note que, para $1 \leq i \leq t$ se tiene que

$$\beta_i + i - s = n + 1 + h_i + i - (t + n + 1) = 0 \tag{1.2}$$

Además, si $i > t$ tenemos que

$$\beta_i + i - s = (n + 1 + t - i) + i - (t + n + 1) = 0 \tag{1.3}$$

por lo que $\phi(h) = \phi(\beta)$. Por el lema anterior, $0 \notin h$ implica que h es la sucesión de longitudes de gancho de la primera columna de $\phi(h) = \phi(\beta)$.

∎

El siguiente corolario nos dice que si tenemos una sucesión β y las longitudes de gancho de la primera columna de $\phi(\beta)$, entonces ambas están relacionadas y ambas generan a la misma partición.

Corolario: Sea $\beta \subset \mathbb{N}_0$ y sean $h = \{h_1 > \ldots > h_t\}$ las longitudes de gancho de la primera columna de $\phi(\beta)$. Entonces $\beta = h$ o existe $n \in \mathbb{N}_0$ tal que

$$\beta = \{n + 1 + h_1, \ldots, n + 1 + h_t, n, \ldots, 1, 0\}.$$

En cualquier caso, $\phi(\beta) = \phi(h)$.

Demostración

Si $0 \notin \beta$, entonces $\beta = h$. Supongamos que $0 \in \beta$. Sea $n \in \mathbb{N}_0$ máximo tal que $n \in \beta$. Escriba

$$\beta = \{n + 1 + \gamma_1 > \ldots > n + 1 + \gamma_{s-1} > n + 1 + \gamma_s, n, \ldots, 1, 0\} \tag{1.4}$$

con $\gamma_i \in \mathbb{Z}^+$ para toda i. Por el lema anterior, $\phi(\gamma) = \phi(\beta)$ y $\gamma = h$.

∎

Veamos ahora algunos ejemplos de β y sus longitudes de gancho de la primera columna de $\phi(\beta)$.

Ejemplos: Las longitudes de gancho de la primera columna de $\phi(\beta)$ son:

1. $\beta = \{6, 5, 2, 1, 0\}$ $\quad h = \{3, 2\}$
2. $\beta = \{10, 8, 7, 0\}$ $\quad h = \{9, 7, 6\}$
3. $\beta = \{9, 7, 5, 4, 2, 1, 0\}$ $\quad h = \{6, 4, 2, 1\}$
4. $\beta = \{11, 10, 7, 5, 4, 3, 2, 1, 0\}$ $\quad h = \{5, 4, 1\}$

En el siguiente lema mostraremos algunas propiedades entre una partición α, sus longitudes de gancho de la primera columna y la partición obtenida al remover un gancho sesgado R^{α}_{ij} de α.

Lema: Sea $\alpha = \{\alpha_1, \ldots, \alpha_r\}$ con $\alpha_r > 0$ una partición con longitudes de gancho de la primera columna $h = \{h_1, \ldots, h_r\}$. Sea (i, j) un nodo de $[\alpha]$, y sea $\lambda = \{\lambda_1, \ldots, \lambda_r\}$ la partición que se obtiene al remover el gancho sesgado R^{α}_{ij} de α. Tenemos que

1. $\lambda_l = \alpha_l$ para toda $1 \leq l \leq i$.
2. Sea $i_0 = i_0(j)$ tal que $i_0 \leq r$ y $\alpha_{i_0+1} < j \leq \alpha_{i_0}$ (donde posiblemente $\alpha_{i_0+1} = 0$). Note que i_0 está bien definida, pues $j \leq \alpha_i$. Entonces $i_0 \geq i$, y $\lambda_l = \alpha_l$ para toda $l > i_0$.
3. $\lambda_l = \alpha_{l+1} - 1$ para toda $i \leq l < i_0$.
4. $\lambda_{i_0} = j - 1 = \alpha_i - i - |R^{\alpha}_{ij}| + i_0$.
5. $h_{i_0+1} < h_i - |R^{\alpha}_{ij}| < h_{i_0}$.
6. Si $q \in \{0, \ldots, h_i - 1\}$ y $q \notin \{|R^{\alpha}_{i1}|, |R^{\alpha}_{i2}|, \ldots, |R^{\alpha}_{i\alpha_j}|\}$, entonces $h_i - q \in h$.

Demostración

1. Note que R^{α}_{ij} no intersecta los primeros $i - 1$ renglones de α.

2. $i_0 \geq i$ pues $j \leq \alpha_i$. Además, el i_0-ésimo es el último renglón de $[\alpha]$ que intersecta a la j-ésima columna y por lo tanto el último renglón de $[\alpha]$ que intersecta a R_{ij}^{α}.

3. Para $i \leq l < i_0$, la parte del l-ésimo renglón de α que intersecta a R_{ij}^{α} consta de los últimos $(\alpha_l - \alpha_{l+1}) + 1$ nodos.

4. Los primeros $j-1$ nodos del renglón i_0 de α son los únicos que no intersectan a R_{ij}^{α}, por lo tanto, $\lambda_{i_0} = j-1$. Sabemos que

$$|R_{ij}^{\alpha}| = |H_{ij}^{\alpha}| = \alpha_i - (j-1) + i_0 - i,$$

así que $j - 1 = \alpha_i - |R_{ij}^{\alpha}| + i_0 - i$.

5. Tenemos que $h_i = \alpha_i + r - i$, $|R_{ij}^{\alpha}| = \alpha_i - (j-1) + i_0 - i$, por lo que

$$h_i - |R_{ij}^{\alpha}| = \alpha_i + r - i - (\alpha_i - j + 1 + i_0 - i) = r + j - 1 - i_0.$$

Además,

$$h_{i_0} = \alpha_{i_0} + r - i_0 \geq j + r - i_0 > j - 1 + r - i_0 = h_i - |R_{ij}^{\alpha}|$$

y

$$h_{i_0+1} = \alpha_{i_0+1} + r - (i_0 + 1) < j + r - i_0 - 1 = h_i - |R_{ij}^{\alpha}|.$$

6. Por la parte (5), $h_i - |R_{ij}^{\alpha}| \notin h$ para toda $1 \leq j \leq \alpha_i$. Equivalentemente, si $H = \{h_i - h_l | l > i\}$ y $R = \{|R_{ij}^{\alpha}| | 1 \leq j \leq \alpha_i\}$, tenemos que $H \cap R = \emptyset$. Pero $H \cup R \subseteq \{1, \ldots, h_i\}$, y $|H| + |R| = r - i + \alpha_i = h_i$, por lo que $\{1, \ldots, h_i\}$ es la unión disjunta de H y R. Como $q \in H \cup R$ y $q \notin R$, entonces $q \in H$, es decir, existe $l > i$ tal que $q = h_i - h_l$, esto es, $h_i - q = h_l \in h$.

∎

El siguiente ejemplo tiene el propósito de aclarar algunos detalles vistos en el teorema.

Ejemplo:

$$
\begin{array}{cccccccccccccccc}
 & & & & j & & & & & & \alpha_i & & & & \\
 & & & & \downarrow & & & & & & \downarrow & & & & \\
 & & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \\
 & & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & & & \\
 & & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & & & \\
i & \longrightarrow & \times & \times & \otimes & \times & \times & \boxtimes & \boxtimes & \boxtimes & \boxtimes & & & & \\
 & & \times & \times & \times & \times & \boxtimes & \boxtimes & & & & & & & R^{\alpha}_{i,j} \\
 & & \times & \times & \times & \boxtimes & \boxtimes & & & & & & & & \\
 & & \times & \times & \boxtimes & \boxtimes & & & & & & & & & \\
 & & \times & \times & \boxtimes & & & & & & & & & & \\
i_0 & \longrightarrow & \times & \times & \boxtimes & & & & & & & & & & \\
 & & \times & \times & & & & & & & & & & & \\
 & & \times & & & & & & & & & & & & \\
 & & \times & & & & & & & & & & & & \\
 & & \times & & & & & & & & & & & &
\end{array}
$$

El siguiente lema nos muestra algunas propiedades entre una sucesión β, su partición α asociada y las longitudes de gancho de la primera columna de α.

Lema: Sea $\alpha = \{\alpha_1, \ldots, \alpha_r \neq 0\}$ una partición. Sea $\beta = \{\beta_1 > \ldots \beta_s\} \subset \mathbb{N}_0$ tal que $\phi(\beta) = \alpha$ y sean $h = \{h_1 > \ldots > h_t\}$ las longitudes de los ganchos de la primera columna de $[\alpha]$. Entonces

1. $\beta_l \geq h_l$ para toda $1 \leq l \leq r$.
2. $|R^{\alpha}_{ij}| \leq \beta_i$ para todo nodo (i, j) de $[\alpha]$.

Demostración

1. Si $0 \notin \beta$ entonces $\beta = h$ y $\beta_l = h_l$ para toda l. Supongamos que $0 \in \beta$. Sea $n \in \mathbb{N}_0$ tal que $\{0, \ldots, n\} \subseteq \beta$ pero $n + 1 \notin \beta$. Entonces, $h_l = \beta_l - (n + 1) \leq \beta_l$ para toda $1 \leq l \leq t$.
2. Por (1) tenemos $h_i \leq \beta_i$. Además $h_i = |R^{\alpha}_{i1}| > |R^{\alpha}_{i2}| > \ldots > |R^{\alpha}_{i\alpha_i}|$. Por lo tanto, $|R^{\alpha}_{ij}| \leq \beta_i$.

$\blacksquare$

Veamos ahora un ejemplo para aclarar el lema anterior.

Ejemplo:

$$
\begin{array}{ccccccccccl}
 & & & j & & & & & & & \\
 & & & \downarrow & & & & & & & \\
i & \longrightarrow & \times & \otimes & \times & \times & \times & \times & \times & & \alpha = (7,5,4,4,2,1,1) \\
 & & \times & \times & \times & \times & \times & & & & \alpha \vdash 24 \\
 & & \times & \times & \times & \times & & & & & h = (13,10,8,7,4,2,1) \\
 & & \times & \times & \times & \times & & & & & \\
 & & \times & \times & & & & & & & \beta = (15,12,10,9,6,4,3,1,0) \\
 & & \times & & & & & & & & \\
 & & \times & & & & & & & & |R^{\alpha}_{1,2}| = 10 \leq \beta_1 = 15
\end{array}
$$

Lema: Sean $\beta = \{\beta_1 > \beta_2 > \ldots > \beta_s\} \subset \mathbb{N}_0, \alpha = \phi(\beta) = \{\alpha_1, \ldots, \alpha_r\}, q \in \mathbb{Z}^+$ y $\beta_k \in \beta$ tal que $0 \leq \beta_k - q \notin \beta$. Entonces $k \leq r$ y existe un único j tal que $|R^{\alpha}_{kj}| = q$.

Demostración

Sean $h = \{h_1 > \ldots > h_r\}$ las longitudes de gancho de la primera columna de α. Supongamos que $\beta \neq h$. Entonces existe $n \in \mathbb{N}_0$ tal que

$$\beta = \{n + 1 + h_1, \ldots, n + 1 + h_r, n, \ldots, 1, 0\},$$

en cuyo caso se debe tener $k \leq t$, pues de lo contrario $\beta_k \in \{0, \ldots, n\}$ y $0 \leq \beta_k - q \notin \beta$ sería imposible. Más aún, se debe tener $n + 1 \leq \beta_k - q = n + 1 + h_k - q$, es decir, $0 \leq h_k - q$. Por otro lado, si $h_k - q = h_l$, entonces $\beta_k - q = n + 1 + h_k - q = n + 1 + h_l = \beta_l$, contradiciendo que $\beta_k - q \notin \beta$. Así que sin pérdida de generalidad, $\beta = h$. En particular, $k \leq r$. Como $|R^{\alpha}_{k1}| > |R^{\alpha}_{k2}| > \ldots$, la unicidad de la j tal que $|R^{\alpha}_{kj}| = q$ está garantizada. Si $q \neq |R^{\alpha}_{kj}|$ para toda j, entonces $h_k - q \in h$, contradicción. Así que existe una única j tal que $|R^{\alpha}_{kj}| = q$.

$\blacksquare$

Ahora probaremos el Teorema Fundamental, la demostración de este teorema es larga y tiene cuatro casos para demostrar la primera parte.

Teorema: TEOREMA FUNDAMENTAL. Sea $\beta = \{\beta_1 > \beta_2 > \ldots > \beta_s\}$ y sea $\alpha = \phi(\beta) = \{\alpha_1, \ldots, \alpha_r\}$

1. Si λ es la partición que se obtiene removiendo el gancho sesgado R^{α}_{ij} de α, entonces β_i es tal que $0 \leq \beta_i - |R^{\alpha}_{ij}| \notin \beta$ y $\phi[(\beta \backslash \{\beta_i\}) \cup \{\beta_i - |R^{\alpha}_{ij}|] = \lambda$

2. Si β_i es tal que $0 \leq \beta_i - q \notin \beta$ para alguna $q \in \mathbb{Z}^+$, entonces $\phi[(\beta \setminus \{\beta_i\}) \cup \{\beta_i - q\}]$ es la partición que se obtiene de α removiendo un $q - gancho$ sesgado R^{α}_{ij} para alguna única j.

Demostración

1. Tenemos que $R^{\alpha}_{ij} \leq \beta_i$. Sean $h = \{h_1 > h_2 > \cdots h_r\}$ las longitudes de gancho de la primera columna de α. Sabemos que existe $i_0 \leq r$ tal que $\beta_\ell = h_\ell + n$ para todo $1 \leq \ell \leq r$. Supongamos que $h = \beta$. Tenemos que existe $i_0 \leq t$ tal que $h_{i_0+1} < h_i - |R^{\alpha}_{ij}| < h_{i_0}$, donde convenimos que $h_{r+1} = 0$. Supongamos ahora que $h \neq \beta$. Sumando n obtenemos: $n + h_{i_0+1} < n + h_i - |R^{\alpha}_{ij}| = \beta_i - |R^{\alpha}_{ij}| < n + h_{i_0} = \beta_{i_0}$. Ahora, si $i_0 \leq r$, entonces $n + h_{i_0+1} = \beta_{i_0+1}$; si $i_0 + 1 = r + 1$, entonces $\beta_{i_0+1} = n - 1 < n = n + h_{i_0+1}$. En cualquier caso tenemos la desigualdad: $\beta_{i_0+1} < \beta_i - |R^{\alpha}_{ij}| < \beta_{i_0}$, por lo que $\beta_i - |R^{\alpha}_{ij}| \notin \beta$. Falta demostrar que $\phi[(\beta \setminus \{\beta_i\}) \cup \{\beta_i - |R^{\alpha}_{ij}|\}] = \lambda$. Sean $\mu = \phi[(\beta \setminus \{\beta_i\}) \cup \{\beta_i - |R^{\alpha}_{ij}|\}] = \{\mu_1 > \mu_2 > \cdots \mu_r\}$, $\lambda = \{\lambda_1 > \lambda_2 > \cdots > \lambda_r\}$. Note que es posible que μ o λ tenga menos de r partes, pero ninguna tiene más de r, (las partes de $\phi(\beta)$ corresponden a los números en $\beta \setminus \{0, \ldots, n\}$).

   Por demostrar: $\lambda_\ell = \mu_\ell$ para todo $1 \leq \ell \leq r$.

   Sea $\gamma = (\beta \setminus \{\beta_i\}) \cup \{\beta_i - |R^{\alpha}_{ij}|\} = \gamma_1 > \gamma_2 > \ldots > \gamma_s$. Observe que $\beta_{i_0+1} < \beta_i - |R^{\alpha}_{ij}| < \beta_{i_0}$, por lo que $\gamma_1 = \beta_1, \gamma_2 = \beta_2, \ldots, \gamma_{i-1} = \beta_{i-1}, \gamma_i = \beta_{i+1}, \gamma_{i+1} = \beta_{i+2}, \ldots, \gamma_{i_0-2} = \beta_{i_0-1}, \gamma_{i_0-1} = \beta_{i_0}, \gamma_{i_0} = \beta_i - |R^{\alpha}_{ij}|, \gamma_{i_0+1} = \beta_{i_0+1}, \gamma_{i_0+2} = \beta_{i_0+2}, \ldots, \gamma_s = \beta_s$.

   Hay cuatro casos:

   Caso 1 $1 \leq \ell < i$. Por un lema anterior, tenemos que $\lambda_\ell = \alpha_\ell = \beta_\ell + \ell - s$. Por otro lado, $\mu_\ell = \gamma_\ell + \ell - s = \beta_\ell + \ell - s = \lambda_\ell$.

   Caso 2 $1 \leq \ell < i_0$. Por un lema anterior, $\lambda_\ell = \alpha_{\ell+1} - 1 = \beta_{\ell+1} + \ell + 1 - s - 1 = \gamma_\ell + \ell + 1 - s - 1 = \gamma_\ell + \ell - s = \mu_\ell$.

   Caso 3 $\ell = i_0$. Por un lema anterior, $\lambda_{i_0} = \alpha_i - i - |R^{\alpha}_{ij}| + i_0$. Por otro lado, $\mu_{i_0} = \gamma_{i_0} + i_0 - s = \beta_i - |R^{\alpha}_{ij}| + i_0 - s = \beta_i + i - i - |R^{\alpha}_{ij}| + i_0 - s = \alpha_i - i - |R^{\alpha}_{ij}| + i_0 = \lambda_{i_0}$.

Caso 4 $i_0 < \ell$. Por un lema anterior, $\lambda_\ell = \alpha_\ell$. Por otro lado, $\mu_\ell = \gamma_\ell + \ell - s = \beta_\ell + \ell - s = \alpha_\ell = \lambda_\ell$.

2. Tenemos que $i \leq r$ y existe un único j tal que $|R^\alpha_{ij}| = q$. Por la parte (1), si λ denota la partición que se obtiene removiendo el gancho sesgado $|R^\alpha_{ij}|$ de α, entonces $\phi[(\beta \setminus \{\beta_i\}) \cup \{\beta_i - q\}] = \lambda$.

∎

Finalmente, para terminar este capítulo, veamos algunos ejemplos de remover ganchos sesgados de las particiones obtenidas a partir de una sucesión β.

Ejemplos: Remover 3-ganchos sesgados de $\phi(\beta)$:

1.

$$\beta = \{\mathbf{9}, 5, 2, 1, 0\} \qquad \alpha = \phi(\beta) = \{5, 2\} \qquad \begin{array}{lllll} \times & \times & \boxtimes & \boxtimes & \boxtimes \\ \times & \times & & & \end{array}$$

$$\beta = \{\mathbf{6}, 5, 2, 1, 0\} \qquad \alpha = \phi(\beta) = \{2, 2\} \qquad \begin{array}{ll} \times & \boxtimes \\ \boxtimes & \boxtimes \end{array}$$

$$\beta = \{5, 3, 2, 1, 0\} \qquad \alpha = \phi(\beta) = \{1\} \qquad \times$$

2.

$$\beta = \{6, 5\} \qquad \alpha = \phi(\beta) = \{5, 5\} \qquad \begin{array}{lllll} \times & \times & \times & \times & \boxtimes \\ \times & \times & \times & \boxtimes & \boxtimes \end{array}$$

$$\beta = \{5, 3\} \qquad \alpha = \phi(\beta) = \{4, 3\} \qquad \begin{array}{llll} \times & \times & \times & \times \\ \boxtimes & \boxtimes & \boxtimes & \end{array}$$

$$\beta = \{5, 0\} \qquad \alpha = \phi(\beta) = \{4\} \qquad \begin{array}{llll} \times & \boxtimes & \boxtimes & \boxtimes \end{array}$$

$$\beta = \{2, 0\} \qquad \alpha = \phi(\beta) = \{1\} \qquad \times$$

Ejemplo: Remover 2-ganchos sesgados de $\phi(\beta)$:

$$\beta = \{6,4,2\} \qquad \alpha = \phi(\beta) = \{4,3,2\} \qquad \begin{array}{cccc} \times & \times & \times & \times \\ \times & \times & \times & \\ \boxtimes & \boxtimes & & \end{array}$$

$$\beta = \{5,3\} \qquad \alpha = \phi(\beta) = \{4,3\} \qquad \begin{array}{cccc} \times & \times & \times & \times \\ \times & \boxtimes & \boxtimes & \end{array}$$

$$\beta = \{5,1\} \qquad \alpha = \phi(\beta) = \{4,1\} \qquad \begin{array}{cccc} \times & \times & \boxtimes & \boxtimes \\ \times & & & \end{array}$$

$$\beta = \{3,1\} \qquad \alpha = \phi(\beta) = \{2,1\} \qquad \begin{array}{cc} \times & \times \\ \times & \end{array}$$

Cabe señalar que remover ganchos sesgados nos lleva a la misma partición sin importar el orden en que se quiten; se demuestra más adelante, usando una herramienta poderosa llamada el ábaco de James.

Capítulo 2

El ábaco de James

En este capítulo definiremos el ábaco de James. El ábaco de James es una herramienta poderosa que nos facilitará el trabajo con las particiones, veremos cómo representar una partición en el ábaco de James, para ello usaremos las longitudes de gancho de la primera columna de la partición α, o una sucesión β. Definiremos también algunas operaciones en el ábaco de James y finalmente encontraremos el q-corazón de una partición, en particular nos interesa el 2-corazón de una partición, con la ayuda del ábaco de James.

Definición:

El q-ábaco de James Considere un ábaco cuyos alambres son q columnas (con $q \in \mathbb{Z}^+$). Numere los q alambres del 0 al $q-1$, de izquierda a derecha.

$$\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ & & & & \\ \odot & \odot & \odot & \odot & \odot \\ \odot & \odot & \odot & \odot & \odot \\ \odot & \odot & \odot & \odot & \odot \\ \odot & \odot & \odot & \odot & \odot \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{array} \qquad\qquad \begin{array}{cc} 0 & 1 \\ & \\ \odot & \odot \\ \odot & \odot \\ \odot & \odot \\ \odot & \odot \\ \vdots & \vdots \end{array}$$

El 5-ábaco de James. El 2-ábaco de James.

> Las posibles posiciones de las cuentas se determinan suponiendo que todas las cuentas están inicialmente arriba y que movemos cada cuenta un ancho de cuenta a la vez. Etiquetamos estas posibles posiciones de las cuentas como sigue:
>
> $$\begin{array}{ccccccc} 0 & 1 & 2 & 3 & \cdots & q-2 & q-1 \\ q & q+1 & q+2 & q+3 & \cdots & 2q-2 & 2q-1 \\ 2q & 2q+1 & 2q+2 & 2q+3 & \cdots & 3q-2 & 3q-1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$
>
> Dado $\beta \subseteq \mathbb{N}_0$ finito, se le asocia una q-configuración de cuentas donde las posiciones de las cuentas determinan los elementos de β.

Veamos algunos ejemplos para familiarizarnos con este nuevo concepto.

Ejemplos:

1. Sean $\beta = \{17, 13, 11, 9, 6, 5, 2, 1, 0\}$ y $q = 3$. La 3-configuración asociada a β es:

   $$\begin{array}{ccc} \odot^{0} & \odot^{1} & \odot^{2} \\ \blacksquare^{3} & \blacksquare^{4} & \odot^{5} \\ \odot^{6} & \blacksquare^{7} & \blacksquare^{8} \\ \odot^{9} & \blacksquare^{10} & \odot^{11} \\ \blacksquare^{12} & \odot^{13} & \blacksquare^{14} \\ \blacksquare^{15} & \blacksquare^{16} & \odot^{17} \end{array}$$

   donde $\odot$ es una cuenta y $\blacksquare$ un espacio vacío. De ahora en adelante, vamos a dibujar las q-configuraciones asociadas a β como sigue:

   $$\begin{array}{ccc} \odot & \odot & \odot \\ \blacksquare & \blacksquare & \odot \\ \odot & \blacksquare & \blacksquare \\ \odot & \blacksquare & \odot \\ \blacksquare & \odot & \blacksquare \\ \blacksquare & \blacksquare & \odot \end{array}$$

2. Sean $\beta = \{2, 4, 5, 6, 7\}$, $q = 2$. La 2-configuración asociada a β es:

$$\begin{array}{cc} \cdot & \cdot \\ \odot & \cdot \\ \odot & \odot \\ \odot & \odot \end{array}$$

Su partición correspondiente es: $\alpha = \phi(\beta) = \{3, 3, 3, 3, 2\}$.

3. Sean $\beta = \{2, 3, 5, 6, 7, 8, 12, 15, 21, 22, 25\}$, $q = 5$. La 5-configuración asociada a β es:

$$\begin{array}{ccccc} \cdot & \cdot & \odot & \odot & \cdot \\ \odot & \odot & \odot & \odot & \cdot \\ \cdot & \cdot & \odot & \cdot & \cdot \\ \odot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \odot & \odot & \cdot & \cdot \\ \odot & \cdot & \cdot & \cdot & \cdot \end{array}$$

4. Sean $\beta = \{0, 1, 3, 4, 8, 9, 12\}$, $q = 2$. La 2-configuración asociada a β es:

$$\begin{array}{cc} \odot & \odot \\ \cdot & \odot \\ \odot & \cdot \\ \cdot & \cdot \\ \odot & \odot \\ \cdot & \cdot \\ \odot & \cdot \end{array}$$

5. Sean $\beta = \{2, 5, 7, 8, 9, 11, 13, 15\}, q = 4$. La 4-configuración asociada a β es:

$$\begin{array}{cccc} \cdot & \cdot & \odot & \cdot \\ \cdot & \odot & \cdot & \odot \\ \odot & \odot & \cdot & \odot \\ \cdot & \odot & \cdot & \odot \end{array}$$

La siguiente proposición nos dirá cómo obtener las longitudes de gancho de la primera columna de la partición α a partir del ábaco de James.

Proposición: Dada una configuración de cuentas, se obtienen las longitudes de gancho de la primera columna del diagrama correspondiente de la siguiente forma: cuente el primer hoyo como cero y empiece a contar a partir de él.

Demostración Esto es una reformulación de un corolario anterior. ▮

Veamos un ejemplo de cómo se hace.

Ejemplo: Para $\beta = \{0, 1, 2, 5, 6\}$, $q = 3$, su configuración de cuentas es:

$$\begin{array}{ccc} \odot^{0} & \odot^{1} & \odot^{2} \\ \cdot^{3}_{0} & \cdot^{4}_{1} & \odot^{5}_{2} \\ \odot^{6}_{3} & \cdot^{7}_{4} & \cdot^{8}_{5} \end{array}$$

Las longitudes de gancho de la primera columna son: $h = \{2, 3\}$

En el ábaco de James nos interesa aprovechar la facilidad con que se pueden realizar operaciones que en el diagrama de la partición pudieran resultar más complicadas. Por ejemplo remover un q-gancho sesgado de una partición es equivalente a deslizar una cuenta un espacio hacia arriba en el ábaco de James, como lo veremos en la siguiente proposición.

Proposición: Sean $\beta \subseteq \mathbb{N}_0$ finito, $\alpha = \phi(\beta)$, $q \in \mathbb{Z}^+$. Entonces remover un q-gancho sesgado a $[\alpha]$ es equivalente a deslizar una cuenta un espacio arriba en la q-configuración de cuentas de β.

Demostración Por el Teorema Fundamental, remover un q-gancho sesgado a $[\alpha]$ es equivalente a cambiar un β_i por $\beta_i - q$, lo cual equivale a deslizar la cuenta correspondiente a la posición β_i un espacio hacia arriba. ▮

Entonces, para encontrar el q-corazón de una partición dado su ábaco de James, es suficiente con recorrer todas las cuentas hacia arriba tanto como se pueda, esto es, tantas veces como podamos quitar q-ganchos sesgados. El resultado será el q-corazón de la partición original y dicho q-corazón es único.

Corolario: Todo diagrama $[\alpha]$ tiene un único q-corazón.

Demostración Sea $\beta \subseteq \mathbb{N}_0$ finito tal que $\alpha = \phi(\beta)$ (por ejemplo, $\beta =$las longitudes de gancho de la primera columna de α). Por la proposición anterior, quitar q-ganchos sesgados equivale a deslizar cuentas en la q-configuración de β. Así, la q-configuración de cuentas de un q-corazón de α ocurre cuando todas las cuentas están tan arriba como sea posible. Esta q-configuración es independiente del orden en el que se deslicen las cuentas. ▮

Los q-corazones que nos interesan son los 2-corazones, y estos se obtienen del 2-ábaco de James, esto es, un ábaco con dos alambres.

Para concluír el capítulo veremos unos ejemplos de particiones para las cuales encontraremos su 2-corazón correspondiente a cada partición usando el diagrama de la partición y usando el 2-ábaco de James.

Ejemplos:

1. $q = 2$, $\alpha = (4, 3, 1)$, $h = (6, 4, 1)$

$$\begin{array}{cccc} \times & \times & \times & \times \\ \times & \boxtimes & \boxtimes & \\ \times & & & \end{array} \dashrightarrow \begin{array}{cccc} \times & \times & \boxtimes & \boxtimes \\ \times & & & \\ \times & & & \end{array} \dashrightarrow \begin{array}{cc} \times & \times \\ \boxtimes & \\ \boxtimes & \end{array} \dashrightarrow \begin{array}{cc} \boxtimes & \boxtimes \end{array}$$

$$\begin{array}{cc} \cdot & \odot \\ \cdot & \cdot \\ \odot & \cdot \\ \odot & \cdot \end{array} \dashrightarrow \begin{array}{cc} \odot & \odot \\ \odot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{array}$$

α tiene 2-corazón vacío.

2. $q = 2$, $\alpha = (5, 2)$, $h = (6, 4)$

$$\begin{array}{ccccc} \times & \times & \times & \boxtimes & \boxtimes \\ \times & \times & & & \end{array} \dashrightarrow \begin{array}{ccc} \times & \times & \times \\ \boxtimes & \boxtimes & \end{array} \dashrightarrow \begin{array}{ccc} \times & \boxtimes & \boxtimes \end{array} \dashrightarrow \times$$

$$\begin{array}{cc} \cdot & \cdot \\ \odot & \cdot \\ \cdot & \cdot \\ \odot & \cdot \end{array} \dashrightarrow \begin{array}{cc} \odot & \cdot \\ \odot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{array}$$

α tiene 2-corazón la partición triangular de tamaño 1.

3. $q = 2$, $\alpha = (4, 3)$, $h = (5, 3)$

$$\begin{array}{cccc} \times & \times & \times & \times \\ \times & \boxtimes & \boxtimes & \end{array} \dashrightarrow \begin{array}{cccc} \times & \times & \boxtimes & \boxtimes \\ \times & & & \end{array} \dashrightarrow \begin{array}{cc} \times & \times \\ \times & \end{array}$$

$$\begin{array}{cc} \cdot & \cdot \\ \cdot & \odot \\ \cdot & \odot \end{array} \dashrightarrow \begin{array}{cc} \cdot & \odot \\ \cdot & \odot \\ \cdot & \cdot \end{array}$$

α tiene 2-corazón la partición triangular de tamaño 2.

4. $q = 2$, $\alpha = (5, 4, 1)$, $h = (7, 5, 1)$

$$\begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & \boxtimes & \boxtimes & \\ \times & & & & \end{array} \dashrightarrow \begin{array}{ccccc} \times & \times & \times & \boxtimes & \boxtimes \\ \times & \times & & & \\ \times & & & & \end{array} \dashrightarrow \begin{array}{ccc} \times & \times & \times \\ \times & \times & \\ \times & & \end{array}$$

$$\begin{array}{cc} \cdot & \odot \\ \cdot & \cdot \\ \cdot & \odot \\ \cdot & \odot \end{array} \dashrightarrow \begin{array}{cc} \cdot & \odot \\ \cdot & \odot \\ \cdot & \odot \\ \cdot & \cdot \end{array}$$

α tiene 2-corazón la partición triangular de tamaño 3.

Capítulo 3

Particiones 2-estables

En este capítulo definiremos las particiones 2-regulares, las particiones 2-estables, enunciaremos algunos resultados para particiones 2-estables, enunciaremos un resultado que dice que si una partición es 2-estable, entonces es 2-regular. Analizaremos la conjetura hecha en el artículo Radha Kessar - Luis Valero Elizondo, Stable partitions and Alperin's weight conjecture for the symmetric groups in characteristic two, Boletín Sociedad Matemática Mexicana (3) Vol. 10, (2004), 53-62 acerca de la existencia de una única matriz de particiones con ciertas características.

Definición: Una partición se dice p**-regular** si las partes de la partición se repiten a lo más $p-1$ veces, esto es, si no hay partes que se repitan más de $p-1$ veces.

Observación: Una partición se dice 2**-regular** si las partes de la partición se repiten a lo más una vez, esto es, si no hay partes repetidas.

Observación: Una partición es 2-regular si no tiene renglones consecutivos del mismo tamaño, que en términos del ábaco quiere decir que no haya cuentas consecutivas.

Veamos algunos ejemplos de particiones p-regulares para entender mejor.

Ejemplos:

1. $\alpha = (5, 4, 2)$ es una partición 2-regular.

$$\begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \\ \times & \times & & & \end{array}$$

2. $\alpha = (5, 3, 3)$ no es una partición 2-regular, ya que se repite el 3.

$$\begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & \times & & \\ \times & \times & \times & & \end{array}$$

3. $\alpha = (7, 5, 4, 2)$ es una partición 2-regular, ya que ningún número se repite.

$$\begin{array}{ccccccc} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & & \\ \times & \times & \times & \times & & & \\ \times & \times & & & & & \end{array}$$

4. $\alpha = (5, 3, 3)$ es una partición p-regular, para $p \geq 3$ ya que el 3 se repite sólamente 2 veces.

$$\begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & \times & & \\ \times & \times & \times & & \end{array}$$

5. $\alpha = (5, 2, 2)$ es una partición 3-regular, ya que el 2 se repite no más de 2 veces.

$$\begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & & & \\ \times & \times & & & \end{array}$$

Ahora si definiremos una de las partes principales del capítulo: las particiones 2-estables.

Definición: Sea α una partición. Decimos que α es **2-estable** si tiene el mismo número de renglones que su 2-corazón.

Veamos algunos ejemplos de particiones que son o no 2-estables y por qué.

Ejemplos:

1. $\alpha = (5, 4, 3)$ es una partición 2-estable, ya que el número de renglones de su 2-corazón que es la partición triangular de tamaño 3 es el mismo número de renglones que α.

$$\begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \\ \times & \times & \times & & \end{array} \qquad \begin{array}{ccc} \times & \times & \times \\ \times & \times & \\ \times & & \end{array}$$

2. $\alpha = (5, 4, 1)$ es una partición 2-estable, ya que el número de renglones de su 2-corazón que es la partición triangular de tamaño 3 es el mismo número de renglones que α.

$$\begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \\ \times & & & & \end{array} \qquad \begin{array}{ccc} \times & \times & \times \\ \times & \times & \\ \times & & \end{array}$$

3. $\alpha = (5, 4, 3, 2)$ no es una partición 2-estable, ya que el número de renglones de su 2-corazón que es la partición triangular de tamaño 3 no es igual al número de renglones de α.

$$\begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \\ \times & \times & \times & & \\ \times & \times & & & \end{array} \qquad \begin{array}{ccc} \times & \times & \times \\ \times & \times & \\ \times & & \\ & & \end{array}$$

4. $\alpha = (4, 3, 2)$ no es una partición 2-estable, ya que el número de renglones de su 2-corazón que es la partición triangular de tamaño 2 no es igual al número de renglones de α.

$$\begin{array}{cccc} \times & \times & \times & \times \\ \times & \times & \times & \\ \times & \times & & \end{array} \qquad \begin{array}{cc} \times & \times \\ \times & \\ & \end{array}$$

Ahora veremos algunos resultados acerca de particiones 2-estables tomados del artículo Radha Kessar - Luis Valero Elizondo, Stable partitions and Alperin's weight conjecture for the symmetric groups in characteristic two, Boletín Sociedad Matemática Mexicana (3) Vol. 10, (2004), 53-62. No veremos las demostraciones pero pueden consultarse en el artículo.

Proposición: Sea $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_t)$ una partición. Las siguientes afirmaciones son equivalentes:

1. λ es 2-estable;
2. $\lambda \equiv 1(\text{mod}2)$ y $\lambda_i \not\equiv \lambda_{i+1}(\text{mod}2)$ para toda $i = 1, \ldots, t-1$.
3. $\lambda_i \equiv t - i + 1(\text{mod}2)$ pra toda $i = 1, \ldots, t$.
4. λ se obtiene de su 2-corazón adjuntando ladrillos horizontales a los renglones no vaciós de su corazón.

De esta proposición se obtiene un corolario interesante, si una partición es 2-estable, entonces es 2-regular.

Corolario: Si λ es 2-estable, entonces también es 2-regular.

Corolario: Si $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_t)$ es 2-estable,entonces la partición dada por $(\lambda_1 + 1, \lambda_2 + 1, \ldots, \lambda_t + 1, 1)$ es también 2-estable.

Lema: Sea $\mu = (\mu_1, \mu_2, \ldots, \mu_s)$ una partición (no necesariamente 2-regular) con 2-corazón $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_k)$. Sea $\Lambda = \{i|\gamma_i \not\equiv \mu_i(\text{mod}2), 1 \leq i \leq k\}$ el conjunto de renglones "disparejos" de μ.Entonces $s \geq k + 2|\Lambda|$.

Teorema: Sea $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_t)$ una partición 2-estable, y sea $\mu = (\mu_1, \mu_2, \ldots, \mu_s)$ una partición 2-regular que contenga a λ. Si$|\mu| = |\lambda| + t + 1$ y el 2-corazón de μ es $(t+1, t, \ldots, 1)$, entonces $\mu = (\lambda_1+1, \lambda_2+1, \ldots, \lambda_t+1, 1)$ y μ es 2-estable.

Lema: Sea λ una particón 2-regular con 2-corazón γ, y sea t el número de renglones de γ. Si λ no es 2-estable, entonces $|\lambda| - |\gamma| \geq t + 1$.

Corolario: Sea n un entero positivo. Entonces existe sólamente un número finito de particiones 2-estables λ tales que $|\lambda| - |\gamma| \leq n$, donde γ es el 2-corazón de λ. En particular, cualquier tabla de particiones que satisfaga la condición 4 de la sección anterior, en cualquier renglón de esa tabla existen sólamente un número finito de particiones 2-estables.

En este artículo también se conjetura la existencia y unicidad de una matriz de particiones que cumpla con lo siguiente:

1. El subgrupo trivial indexa un renglón que consiste de todas las particiones triangulares (incluimos aquí las particiones triangulares de tamaños 0 y 1 por completez)

2. Cada subgrupo peso Q aparece por primera vez dentro de un grupo simétrico S_n donde n es tal que Q no tiene puntos fijos en el conjunto $\{1, \ldots, n\}$.

   Más aún, si una partición 2-regular de tamaño m aparece en un renglón indexado por el grupo Q, entonces Q es un subgrupo peso de S_m.

3. La primera partición de todo renglón tiene 2-corazón vacío. La segunda partición tiene 2-corazón de tamaño 1, la tercera tiene 2-corazón de tamaño 3 y la cuarta tiene 2-corazón de tamaño 6. En otras palabras, el 2-corazón de cada partición a lo largo de la i-ésima columna es la i-ésima partición triangular (donde $\emptyset$) es la primera partición triangular.

4. Para cada renglón, todas las particiones λ en ese renglón son tales que la diferencia de el tamaño de λ menos el tamaño de su 2-corazón es constante.

5. A lo largo de cada renglón, cada partición está contenida en la que está a su derecha.

La primer afirmación es sólamente el bien conocido hecho de que en característica 2,los únicos grupos simétricos con módulos simples y proyectivos son los S_t con t un número triangular, y que tales módulos son parametrizados por las correspondientes particiones triangulares. El segundo enunciado es probado implícitamente en [Alperin and Fong, 1990]. Es más o menos directo que se ocurra la siguiente conjetura:

Conjetura: Es posible arreglar todas las particiones 2-regulares en una única matriz infinita que satisface las cinco condiciones mencionadas arriba.

Note que la existencia de una matriz infinita de particiones que satisfaga las condiciones 1, 2, 3 y 4 es equivalente a la versión de bloques de la conjetura de Alperin para los grupos simétricos en característica dos. De hecho, todo lo que tenemos que hacer es elegir biyecciones arbitrarias entre los pesos (o, mejor dicho, subgrupos peso) y los irreducibles en sus bloques, los cuales son parametrizados por particiones con 2-corazones apropiados.

Capítulo 4

Resultados originales

En este capítulo veremos la parte principal de la tesis, los resultados originales, también se corrieron algunas rutinas en Gap para comprobar algunas cosas, estas rutinas se verán en el siguiente capítulo.

También veremos algunas observaciones y conjeturas hechas a partir de las observaciones realizadas después de hacer muchos ejemplos de particiones y ábacos.

El objetivo era intentar probar la existencia y unicidad de la matriz infinita de particiones 2-regulares mencionada en el capítulo anterior, para ello, demostré tres proposiciones:

1. La primera proposición nos dice que 2 particiones distintas con 2-corazón vacío pueden estar contenidas en una misma partición con 2-corazón la partición triangular de tamaño uno. Lo que da pie a la idea de que la manera de acomodar las particiones en la matriz de particiones 2-regulares no es única.
2. La segunda proposición tiene la motivación de mostrar la existencia de la matriz por partes, lo ideal sería poder generalizar esta proposición para cualquier partición que tenga 2-corazón la partición triangular de tamaño n. La proposición dice así: Para cualquier partición 2-regular de 2-corazón vacío existe una partición 2-regular de 2-corazón la partición triangular de tamaño uno que la contiene con exactamente un nodo más.

3. Finalmente, la tercera y más importante proposición que veremos en este capítulo es un contraejemplo que prueba que la matriz de particiones 2-regulares que cumple con las propiedades vistas en el capítulo anterior, no es única.

Ahora veamos las primeras dos proposiciones:

Proposición: Sean P_1, P_2, Q_1, Q_2 particiones tales que tienen las siguientes propiedades:

- Las P tienen 2-corazón vacío.
- Las Q tienen 2-corazón la partición triangular de tamaño 1.
- Tanto Q_1 como Q_2 contienen a P_1 y a P_2.
- $P_1 \neq P_2$.

Entonces $Q_1 = Q_2$.

Demostración Supongamos que existen P_1 y P_2 particiones, $P_1 \neq P_2$ tales que P_1 y P_2 tienen 2-corazón vacío y que existen particiones Q_1, Q_2 con 2-corazón la partición triangular de tamaño uno, tales que $P_1, P_2 \subseteq Q_1$ y $P_1, P_2 \subseteq Q_2$.

Por demostrar que $Q_1 = Q_2$.

Tenemos que P_1 se obtiene de Q_1 al quitar un nodo y P_2 se obtiene de Q_1 al quitar un nodo distinto del de P_1 pues $P_1 \neq P_2$. Por tanto, si agregamos un nodo a P_1, obtenemos Q_1 y si agregamos un nodo distinto a P_2, obtenemos Q_1. Esto es, si definimos $P_1 \cup P_2$ como la partición más pequeña que contiene a P_1 y a P_2, tenemos que $Q_1 = P_1 \cup P_2$. De manera similar, Q_2 es la mínima partición que contiene a P_1 y a P_2. Por lo tanto $Q_2 = P_1 \cup P_2$. Por tanto $Q_1 = Q_2$. ∎

Proposición: Para cualquier partición 2-regular de 2-corazón vacío existe una partición 2-regular de 2-corazón la partición triangular de tamaño uno que la contiene con exactamente un nodo más.

Demostración El ábaco de una partición con 2-corazón vacío se puede ver de dos formas:

1. Si tiene un número par de cuentas, entonces hay el mismo número de cuentas en los rieles.
2. Si tiene un número impar de cuentas, entonces hay una cuenta más en el riel izquierdo que en el derecho.

1. Si tiene un número par de cuentas, entonces hay el mismo número de cuentas en los dos rieles. Así que, para pasar a una partición que la contenga y que tenga 2-corazón la partición triangular de tamaño uno, hay que mover una cuenta del riel de la derecha, al riel de la izquierda.
   - Si la última cuenta del ábaco se encuentra en el riel de la derecha, entonces movemos la última cuenta un lugar (esto es, si estaba en el lugar n, ahora estará en el lugar $n + 1$) para que quede en el riel de la izquierda.
   - Si la última cuenta del ábaco se encuentra en el riel de la izquierda, entonces tomamos la última cuenta del riel derecho y la recorremos un lugar a la derecha para que quede en el riel izquierdo.
2. Si tiene un número impar de cuentas, entonces hay exactamente una cuenta más en el riel izquierdo que en el derecho. Así que para pasar a una partición que la contenga y que tenga 2-corazón la partición triangular de tamaño uno, hay que pasar una cuenta del riel izquierdo al derecho.
   - Si la última cuenta está en el riel izquierdo, entonces movemos la última cuenta un lugar a la derecha para que vaya del riel izquierdo al derecho.
   - Si la última cuenta está en el riel derecho, entonces tomamos la última cuenta del riel izquierdo y la movemos un lugar a la derecha, de esta manera, va del riel izquierdo, al riel derecho.

∎

La siguientes observaciónes nos explican cómo construír la siguiente partición usando el ábaco de James de acuerdo a la partición que tengamos.

Observación: Si el diagrama de la partición de 2-corazón vacío consta de un renglón, entonces el 2-ábaco de James tiene sólo una cuenta y ésta se encuentra en el alambre cero.

Para construír la siguiente partición (esto es, una partición que contenga a la primera y con 2-corazón la partición triangular de tamaño uno), es suficiente con recorrer la cuenta en el mismo renglón al alambre uno.

Ahora esta partición es estable, entonces para construír la siguiente partición en el renglón basta con:

1. Recorrer ambos alambres un espacio hacia abajo.
2. En el alambre cero, poner un espacio vacío.
3. En el alambre uno, poner una cuenta.

Observación: Si el diagrama de la partición de 2-corazón vacío consta de dos renglones, entonces el 2-ábaco de James tiene 2 cuentas, una en el alambre cero y la otra en el alambre uno.

Para construír la siguiente partición (esto es, una partición que contenga a la primera y con 2-corazón la partición triangular de tamaño uno), se toma la cuenta que está en el alambre uno, supongamos que está en el lugar p, la ponemos en el lugar $p + 1$ que está en el alambre cero.

Notemos que ahora las dos cuentas están en el alambre cero.

Para construír la siguiente partición (esto es, una partición que contenga a la partición anterior y con 2-corazón la partición triangular de tamaño 2), se toma la primera cuenta y a partir de ella se recorren todas las cuentas y espacios siguientes un lugar (incluyendo la primera cuenta). Esto es, si la primera cuenta está en el lugar p, a partir de ella todos los lugares serán $p + 1, p + 2, \ldots$ Notemos que ahora las dos cuentas están en el alambre uno. Ahora esta partición es estable, así que para construír las siguientes particiones basta con:

1. Recorrer ambos alambres un espacio hacia abajo.
2. Poner una cuenta en el alambre uno.

Después de realizar varios ejemplos usando los diagramas de las particiones y los ábacos de James se me ocurrieron las siguientes conjeturas:

Conjetura: Para pasar de una partición de 2-corazón vacío a una de 2-corazón la partición triangular de tamaño uno, se puede hacer sin agregar un nuevo renglón (excepto por la partición vacía).

Conjetura: Si tengo una partición con n renglones, entonces se puede ir estabilizando sin aumentar renglones en los primeros n pasos.

Veamos unas últimas observaciones acerca de las particiones y los ábacos antes de ver el contraejemplo estrella que nos prueba la tercera proposición de este capítulo.

Observación: Si el diagrama de la partición de 2-corazón vacío consta de 3 renglones, entonces el 2-ábaco de James tiene 3 cuentas: dos en el alambre cero y una en el alambre uno.

Para construír las siguente partición (que contenga a la anterior y con 2-corazón la partición triangular de tamaño uno), hay que pasar la segunda cuenta del alambre cero al alambre uno en el mismo renglón. Notemos que ahora tenemos 2 cuentas en el alambre uno y una cuenta en el alambre cero.

Para construír la siguiente partición (que contenga a la anterior y con 2-corazón la partición triangular de tamaño 2), hay que mover las dos cuentas del alambre uno al alambre cero sumándoles un lugar, esto es, si había una cuenta en el alambre uno en el lugar p, ahora estará en el lugar $p + 1$ en el alambre cero. Notemos que ahora las 3 cuentas están en el alambre cero.

Observación: Si las particiones son 2-corazones, entonces si α, β son particiones y $\alpha \subset \beta$ entonces el 2-ábaco de James de α está contenido en el 2-ábaco de James de β.

Observación: Sean dos particiones α y β. Si el número de espacios de α es menor o igual que el número de espacios de β y el número de cuentas de α es menor o igual que el número de cuentas de β y α está "contenida" en β, entonces el 2-ábaco de James de α está contenido en el 2-ábaco de James de β.

Al final del capítulo anterior, hablamos de una conjetura hecha en el artículo de Radha Kessar - Luis Valero Elizondo, titulado: Stable partitions and Alperin's weight conjecture for the symmetric groups in characteristic two, donde se conjetura la existencia y unicidad de una matriz infinita de particiones 2-regulares que cumple con las 5 condiciones mencionadas en el capítulo anterior.

Después de analizar varios ejemplos a mano, encontré un contraejemplo para la parte de unicidad. Posteriormente, se corrieron en la computadora algunas rutinas para ver hasta qué punto la matriz no es única y vimos que parece ser que la matriz existe, pero es un hecho que está muy lejos de ser única.

A continuación veremos un contraejemplo de que la matriz de la que hablamos no es única, tomemos por ejemplo las particiones 2-regulares del 10 que tienen 2-corazón vacío y las particiones 2-regulares del 11 que tienen 2-corazón la partición triangular de tamaño uno. La cardinalidad de los dos conjuntos es la misma, y Alperin lo asegura. La conjetura dice que existe una única manera de asociar ambos conjuntos de manera que cumplan con las condiciones, pero en este ejemplo, existen 2 maneras diferentes de asociarlos.

Proposición: La matriz infinita de particiones 2-regulares mencionada en el capítulo anterior, no es única.

Demostración Existen 2 maneras diferentes de asociar las particiones 2-regulares del 10 con las particiones 2-regulares del 11.

$$\begin{array}{ccc} (10) & \longrightarrow & (11) \\ (9,1) & \longrightarrow & (9,2) \\ (8,2) & \longrightarrow & (8,2,1) \\ (5,3,2) & \longrightarrow & (5,4,2) \\ (6,3,1) & \longrightarrow & (6,4,1) \\ (6,4) & \longrightarrow & (7,4) \\ (7,3) & \longrightarrow & (7,3,1) \end{array} \qquad \begin{array}{ccc} (10) & \longrightarrow & (11) \\ (9,1) & \longrightarrow & (9,2) \\ (8,2) & \longrightarrow & (8,2,1) \\ (5,3,2) & \longrightarrow & (5,4,2) \\ (6,3,1) & \longrightarrow & (7,3,1) \\ (6,4) & \longrightarrow & (6,4,1) \\ (7,3) & \longrightarrow & (7,4) \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times \\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times \\ \times\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times\ \times \\ \times\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times \\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times\ \times \\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times \\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times \\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times \\ \times\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times\ \times \\ \times\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times \\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times \\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times\ \times \\ \times \end{array}$$

$$\begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times \end{array} \longrightarrow \begin{array}{l} \times\ \times\ \times\ \times\ \times\ \times\ \times \\ \times\ \times\ \times\ \times \end{array}$$

▮

Para particiones 2-regulares más grandes existen bastantes más de dos maneras diferentes de asociarlos y por supuesto, eso incrementa aún más el número de matrices distintas.

Capítulo 5

Código en Gap

En este capítulo veremos una biblioteca de rutinas programadas en Gap que permiten hacer operaciones con las particiones y los ábacos. Veremos primero un índice de las funciones con la explicación de qué es lo que hacen y después veremos los códigos de las rutinas.

Posteriormente veremos las rutinas que junto con las primeras, fueron usadas para encontrar la matriz de particiones, que además, como pudimos comprobar, está muy lejos de ser única. Sin embargo, la existencia no se ha podido refutar.

Finalmente, veremos algunas corridas.

5.1. Índice de funciones

- `particion_n := function(alpha)` Rutina que dada una partición α, dice cuántos nodos tiene la partición. (α es partición de n).
- `renglon_moi_j:= function(part,j)` Rutina que encuentra el último renglón de la partición *part* tal que $part[k] >= j$.
- `traspuesta_particion := function(part)` Rutina que regresa la traspuesta de la partición *part*.
- `caract := function(lg,cont)` Rutina que regresa `0` si cont está en lg y . si no

- `dibuja_abaco := function(lg)` Rutina que dibuja el ábaco correspondiente a la lista de longitudes de gancho de la primera columna lg.

- `encuentra_abaco := function(alpha)` Rutina que dada una partición α, encuentra las longitudes de gancho de la primera columna.

- `encuentra_particion := function(beta)` Rutina que dadas las longitudes de gancho de la primera columna de la partición α, encuentra la partición α.

- `particion_triangular := function(n)` Rutina que calcula la partición triangular n-ésima.

- `encuentra_2_corazon := function(lg)` Rutina que dadas las longitudes de gancho de la primera columna de la partición α, encuentra el 2-corazón de la partición.

- `alpha_2_contiene_alpha_1 := function(alpha_1,alpha_2)` Rutina que dadas dos particiones, α_1 y α_2, decide si α_2 contiene a α_1 y entonces regresa $true$, si α_2 no contiene a α_1 regresa $false$.

- `2corazon_a2contiene2corazon_a1 := function(alpha_1,alpha_2)` Rutina que dadas dos particiones, α_1 y α_2, decide si el 2-corazón de α_2 contiene al 2-corazón de α_1 y en ese caso, regresa $true$, si no, regresa $false$.

- `diferencia_de_cuentas:= function(alpha_1,alpha_2)` Rutina que dadas dos particiones α_1 y α_2, decide si la diferencia de cuentas de los 2-corazones es igual que la diferencia de cuentas de las particiones.

- `diferencia_de_nodos:= function(alpha_1,alpha_2)` Rutina que dadas dos particiones α_1 y α_2, dice si la diferencia de nodos de los 2-corazones es igual que la diferencia de nodos de las particiones.

- `particiones_2_regulares := function(n)` Rutina que encuentra todas las particiones 2-regulares de n.

- `particiones_de_m_con_2_corazon_n:= function(m,n)` Rutina que da todas las particiones de tamaño m que tienen 2-corazón el n-ésimo triángulo.

5.2. Código

```
# relacion_contenciones := function(m,n)
# Rutina que recibe dos parámetros: m y n.
# Queremos las particiones de m que tienen 2-corazón el
# n-ésimo triángulo y las particiones de m+n+1 que tienen
# 2-corazón el n+1-ésimo triángulo.
# Entonces esta rutina construye una matriz que se llama
# relaciones la cual me indica cómo están contenidas las
# particiones de m en las de m+n+1.

# esta_contenida := function(p,q)
# Rutina que recibe dos particiones p y q y regresa true si
# p está contenida en q y false si no.

# contenciones := function(P,Q)
# Rutina que recibe dos listas de particiones P y Q y
# calcula las contenciones de las particiones de P en
# las particiones de Q.

# dibujar_correspondencia := function(listap, listaq,
#                                    correspondencia)
# Rutina que dibuja la correspondencia.

# agrandar_renglon := function(row,max_col,matrix)
# Rutina que agranda un renglón en la matriz matrix.

# agrandar_matriz := function(max_row,max_col,matrix)
# Rutina que agranda la matriz matrix.

# correspondencias := function(relaciones)
# Rutina que recibe la matriz de relaciones y encuentra
# todas las posibles biyecciones entre las particiones.

# f := function(relaciones, listap, listaq, correspondencia,
               resultados)
# Rutina que recibe la matriz de relaciones y
```

```
# -------------------------------------------------------

particion_n := function(alpha)

# Dada una partición alpha, diga cuántos nodos tiene
# la partición. (alpha es partición de n).

  local n, m, i;
  m:=Length(alpha);
  n:=0;
  for i in [1..m] do
    n:=n+alpha[i];
  od;
  return n;
end;

# -------------------------------------------------------

renglon_moi_j:= function(part,j)

# Encuentra el último renglón de la partición
# part tal que part[k]>=j.

local i, n;
  n:=Length(part);
  for i in [1..n] do
    if (part[i]<j) then
      return i-1;
      break;
    elif i=n then
      return n;
      break;
    fi;
  od;
end;

# -------------------------------------------------------
```

```
traspuesta_particion := function(part)

# Rutina que regresa la traspuesta de la partición part.

  local i, m, ptras;
  m:= part[1];
  ptras:= [];
  for i in [1..m] do
    ptras[i]:=renglon_moi_j(part,i);
  od;
  return ptras;
end;

# ------------------------------------------------------

caract := function(lg,cont)

# Rutina que regresa O si cont está en lg y . si no

  if cont in lg then
    return "O";
  else
    return ".";
  fi;
end;

# ------------------------------------------------------

dibuja_abaco := function(lg)

# Rutina que dibuja el ábaco correspondiente a la lista de
# longitudes de gancho de la primera columna lg.

  local n, cont;
  Sort(lg);
  n:=lg[Length(lg)];
  cont:=0;
  while cont<=n do
    Print(caract(lg,cont), "  ", caract(lg, cont+1), "\n");
```

```
    cont:= cont+2;
  od;
  Print("\n \n");
end;

# -------------------------------------------------------

encuentra_abaco := function(alpha)

# Dada una partición, encuentra las longitudes de
# gancho de la primera columna

  local i, n, beta;
  beta:=[];
  n:=Length(alpha);
  for i in [1..n] do
    beta[i]:= alpha[i]+n-i;
  od;
  return beta;
end;

# ------------------------------------------------------

encuentra_particion := function(beta)

# Dadas las longitudes de gancho de la primera
# columna de la partición alpha, encuentra la
# partición alpha.

  local i, n, alpha;
  alpha:=[];
  Sort(beta);
  beta:=Reversed(beta);
  n:=Length(beta);
  for i in [1..n] do
    alpha[i]:=beta[i]-n+i;
  od;
  return alpha;
end;
```

```
# ------------------------------------------------------

particion_triangular := function(n)

# Calcula la partición triangular n-ésima.

  if (n=0) then
    return [];
  else
    return [1..n];
  fi;
end;

# ------------------------------------------------------

encuentra_2_corazon := function(lg)

# Dadas las longitudes de gancho de la primera
# columna de la partición alpha, encuentra
# el 2-corazón de la partición.

  local pares, impares, i, n;
  pares:=0;
  impares:=0;
  n:=Length(lg);
  for i in [1..n] do
    if IsEvenInt(lg[i]) then
      pares:=pares+1;
    else
      impares:=impares+1;
    fi;
   od;
   if (impares>=pares) then
     return particion_triangular(impares-pares);
   else
     return particion_triangular((pares-impares)-1);
   fi;
end;

# -----------------------------------------------------
```

```
alpha_2_contiene_alpha_1 := function(alpha_1,alpha_2)

# Dadas dos particiones, alpha_1 y alpha_2, decide si
# alpha_2 contiene a alpha_1 y entonces regresa true,
# si alpha_2 no contiene a alpha_1 regresa false.

  local i, n, m;
  Sort(alpha_1);
  Sort(alpha_2);
  alpha_1:=Reversed(alpha_1);
  alpha_2:=Reversed(alpha_2);
  n:=Length(alpha_1);
  m:=Length(alpha_2);
  if (n<=m) then
    for i in [1..n] do
      if (alpha_1[i] <= alpha_2[i]) then
        i:=i+1;
      else
        return 0;
        break;
      fi;
    od;
    return true;
  else
    return false;
  fi;
end;

# ---------------------------------------------------

2corazon_a2contiene2corazon_a1 := function(alpha_1,alpha_2)

# Dadas dos particiones, alpha_1 y alpha_2, decide
# si el 2-corazón de alpha_2 contiene al 2-corazón
# de alpha_1 y en ese caso, regresa true, si no,
# regresa false.

  local n, m, beta_1, beta_2, 2_corazon_alpha_1,
        2_corazon_alpha_2;
```

```
  beta_1:=[];
  beta_2:=[];
  2_corazon_alpha_1:=[];
  2_corazon_alpha_2:=[];
  Sort(alpha_1);
  alpha_1:=Reversed(alpha_1);
  Sort(alpha_2);
  alpha_2:=Reversed(alpha_2);
  beta_1:=encuentra_abaco(alpha_1);
  beta_2:=encuentra_abaco(alpha_2);
  2_corazon_alpha_1:=encuentra_2_corazon(beta_1);
  2_corazon_alpha_2:=encuentra_2_corazon(beta_2);
  n:=Length(2_corazon_alpha_1);
  m:=Length(2_corazon_alpha_2);
  if (n<=m) then
    return true;
  else
    return false;
  fi;
end;

# ------------------------------------------------------

diferencia_de_cuentas:= function(alpha_1,alpha_2)

# Dadas dos particiones alpha_1 y alpha_2, diga si
# la diferencia de cuentas de los 2-corazones es igual
# que la diferencia de cuentas de las particiones.

  local n, m, p, q, k, r, beta_1, beta_2, 2_corazon_1,
        2_corazon_2;
  n:=Length(alpha_1);
  m:=Length(alpha_2);
  k:=AbsInt(m-n);
  beta_1:=[];
  beta_2:=[];
  2_corazon_1:=[];
  2_corazon_2:=[];
  beta_1:=encuentra_abaco(alpha_1);
  beta_2:=encuentra_abaco(alpha_2);
```

```
  2_corazon_1:=encuentra_2_corazon(beta_1);
  2_corazon_2:=encuentra_2_corazon(beta_2);
  p:=Length(2_corazon_1);
  q:=Length(2_corazon_2);
  r:=AbsInt(q-p);
  if (k=r) then
    return true;
  else
    return false;
  fi;
end;

# ---------------------------------------------------

diferencia_de_nodos:= function(alpha_1,alpha_2)

# Dadas dos particiones alpha_1 y alpha_2, diga si
# la diferencia de nodos de los 2-corazones es igual
# que la diferencia de nodos de las particiones.

  local n, m, p, q, k, r, beta_1, beta_2, 2_corazon_1,
        2_corazon_2;
  n:=particion_n(alpha_1);
  m:=particion_n(alpha_2);
  k:=AbsInt(m-n);
  beta_1:=[];
  beta_2:=[];
  beta_1:=encuentra_abaco(alpha_1);
  beta_2:=encuentra_abaco(alpha_2);
  2_corazon_1:=[];
  2_corazon_2:=[];
  2_corazon_1:=encuentra_2_corazon(beta_1);
  2_corazon_2:=encuentra_2_corazon(beta_2);
  p:=particion_n(2_corazon_1);
  q:=particion_n(2_corazon_2);
  r:=AbsInt(q-p);
  if (k=r) then
    return true;
  else
    return false;
```

```
  fi;
end;

# ----------------------------------------------------

lista_de_particiones_2_regulares := [];

particiones_2_regulares := function(n)
  local resultado, m, particiones_de_m, p, tmp;
  resultado := [[n]];
  for m in [1..n-1] do
    if ( not IsBound(lista_de_particiones_2_regulares[m]) ) then
      lista_de_particiones_2_regulares[m] :=
                               particiones_2_regulares(m);
    fi;
    particiones_de_m := lista_de_particiones_2_regulares[m];
    for p in particiones_de_m do
      # Checa si es 2-regular
      if ( n-m > p[1] ) then
        Add(resultado,Concatenation([n-m],p));
      fi;
    od;
  od;
  return resultado;
end;

# -----------------------------------------------------

particiones_de_m_con_2_corazon_n:= function(m,n)

# Rutina que da todas las particiones de tamaño m
# que tienen 2-corazón el n-ésimo triángulo.

  local particiones, part, 2_corazones, r, i, beta,
        particiones_m, long_2_corazon, p;
  particiones:=[];
  part:=[];
  2_corazones:=[];
  beta:=[];
  particiones_m:=[];
```

```
  long_2_corazon:=[];
  particiones := particiones_2_regulares(m);
#  particiones:=Partitions(m);
  r:=Length(particiones);
  for i in [1..r] do
    beta[i]:=encuentra_abaco(particiones[i]);
    2_corazones[i]:=encuentra_2_corazon(beta[i]);
    long_2_corazon[i]:=Length(2_corazones[i]);
    if (long_2_corazon[i]=n) then
      Add(part,particiones[i]);
    fi;
  od;
  return part;
end;

# ---------------------------------------------------

relacion_contenciones := function(m,n)

# Rutina que recibe dos parámetros: m y n.
# Queremos las particiones de m que tienen 2-corazón
# el n-ésimo triángulo y las particiones de m+n+1
# que tienen 2-corazón el n+1-ésimo triángulo.
# Entonces esta rutina construye una matriz que se
# llama relaciones la cual me indica cómo están
# contenidas las particiones de m en las de m+n+1.

  local p, q, i, j, relaciones;
  p := [particiones_de_m_con_2_corazon_n(m,n)];
  q := [particiones_de_m_con_2_corazon_n(m+n+1,n+1)];
  relaciones := [];
  for i in [1..Size(p)] do
    relaciones[i] := [];
    for j in [1..Size(q)] do
      if alpha_2_contiene_alpha_1(p[i],q[j]) then
        Add(relaciones[i],j);
      fi;
    od;
  od;
  return relaciones;
```

```
end;


# ------------------------------------------------------

esta_contenida := function(p,q)

# Rutina que recibe dos particiones p y q y regresa
# true si p está contenida en q y false si no.

  local i;
  if (Size(p)>Size(q)) then
    return false;
  fi;
  for i in [1..Size(p)] do
    if (p[i]>q[i]) then
      return false;
    fi;
  od;
  return true;

end;

# -------------------------------------------------------

calcula_contenciones := function(P,Q)

# Rutina que recibe dos listas de particiones P y Q
# y calcula las contenciones de las particiones de
# P en las particiones de Q.

  local p, todas, j, esta_p;
  todas := [];
  for p in P do
    esta_p := [];
    for j in [1..Size(Q)] do
      if (esta_contenida(p,Q[j])) then
        Add(esta_p,j);
      fi;
    od;
```

```
    Add(todas,esta_p);
  od;
  return todas;
end;

# ----------------------------------------------------

dibujar_correspondencia := function(listap, listaq,
                                    correspondencia)

# Rutina que dibuja la correspondencia.

  local i, dibuja_2_particiones;

  dibuja_2_particiones := function(p,q)
    local i,max,col1, col2, caja, separador;
    caja:="[]";
    separador:=" | ";
    for i in [1..Maximum(Size(p),Size(q))] do
      # First column
      if i <= Size(p) then
        col1 := Concatenation(ReplacedString(String("",p[i]),
                " ",caja), String("",(p[1]-p[i])*Size(caja)) );
      else
        col1 := String("", p[1]*Size(caja));
      fi;
      # Second column
      if i <= Size(q) then
        col2 := Concatenation(ReplacedString(String("",q[i]),
                " ",caja), String("",(q[1]-q[i])*Size(caja)) );
      else
        col2 := String("", q[1]*Size(caja));
      fi;
      Print(col1, separador, col2,"\n");
    od;
    Print("\n");
  end;

  for i in [1..Size(listap)] do
    dibuja_2_particiones(listap[i], listaq[correspondencia[i]]);
```

```
  od;
end;

# -----------------------------------------------------

correspondencias := function(relaciones)

# Rutina que recibe la matriz de relaciones y encuentra
# todas las posibles biyecciones entre las particiones.

  local f, listap, listaq, correspondencia, resultados;
  listap := [1..Size(relaciones)];
  listaq := [1..Size(relaciones)];
  correspondencia := [];
  resultados := [];

  f := function(relaciones, listap, listaq, correspondencia,
               resultados)

    local i,j, miselec, mielec, elec, modif, listavp, listavq,
          relacionesv, correspondenciav, correspondencias, id;
    id := Random([1..100]);
    elec := 1;
    while Size(listap)>0 do
      modif := false;
      for i in listap do
        miselec := 0;
        for j in listaq do
          if j in relaciones[i] then
            miselec := miselec + 1;
            mielec := j;
          fi;
        od;
        if miselec=0 then
          if ( Size(resultados) > 0 ) then
            Print("\nNo puedo calcular esta correspondencia!!!\n");
          else
            Print("\nNo existe una correspondencia!!!
                   Lo siento mucho Luis!!!\n");
          fi;
```

```
    listap := [];
    listaq := [];
    #correspondencia := "No pude calcular esta
                           correspondencia!!!";
    #AddSet(resultados, correspondencia);
    return;
  fi;
  if miselec=1 then
    correspondencia[i] := mielec;
    RemoveSet(listap,i);
    RemoveSet(listaq, mielec);
    #Print("id =",id,"\n");
    #Display(correspondencia);
    #Display(listap);
    #Display(listaq);
    modif := true;
    break;
  fi;
  if miselec = elec then
    listavp := ShallowCopy(listap);
    listavq := ShallowCopy(listaq);
    correspondenciav := ShallowCopy(correspondencia);
    relacionesv := ShallowCopy(relaciones);
    correspondencias := [];
    for j in listavq do
      if j in relacionesv[i] then
        listap := ShallowCopy(listavp);
        listaq := ShallowCopy(listavq);
        correspondencia := ShallowCopy(correspondenciav);
        relaciones := ShallowCopy(relacionesv);
        correspondencia[i] := j;
        RemoveSet(listap, i);
        RemoveSet(listaq, j);
        if Size(listap) = 0 then
          AddSet(resultados, correspondencia);
        else
          f(relaciones, listap, listaq, correspondencia,
            resultados);
        fi;
      fi;
```

```
          od;
          return;
        fi;
      od;
      if not modif then
        #Print("Existe más de una manera de hacer la
                              correspondencia!!!\n");
        elec := elec + 1;
      else
        elec := 1;
      fi;
    od;
    AddSet(resultados, correspondencia);
    return;
  end;

  f(relaciones, listap, listaq, correspondencia, resultados);
  return resultados;
end;


# ---------------------------------------------------

agrandar_renglon := function(row,max_col,matrix)

# Rutina que agranda un renglón en la matriz matrix.

  local P, Q, contenciones, tmp, col, sigma, print;
  for col in [0..max_col] do
    if( not IsBound(matrix[row]) ) then
      matrix[row] := [];
    fi;
    if(not IsBound(matrix[row][col+1])) then
      if (col=0) then
        matrix[row][col+1] :=
             [ particiones_de_m_con_2_corazon_n(row,col) ];
        Print("\n\n\n Nueva entrada en matrix: \n ");
        Print("matrix[",row,"][",col+1,"]:=",
                              matrix[row][col+1],"\n#\n");
      else
```

```
        Q :=
      particiones_de_m_con_2_corazon_n(row+col*(col+1)/2,col);
        P := matrix[row][col][1];
        #Print("col := ",col,";\n");
        #Print("P := ",P,";\n");
        #Print("Q := ",Q,";\n");
        contenciones := calcula_contenciones(P,Q);
        #Print("contenciones := ",contenciones,";\n");
        tmp := correspondencias(contenciones);
        matrix[row][col+1] := [];
        Print("\n\n\n Nueva entrada en matrix: ");
        Print("[",row,"][",col+1,"] tiene tamaño ",Size(tmp),"\n#\n");
        print := 1;
        for sigma in tmp do
          Add(matrix[row][col+1],Q{sigma});
          if ( print > 0 ) then
            Print("matrix[",row,"][",col+1,"]:=",
                                     matrix[row][col+1],"\n#\n");
            dibujar_correspondencia(P,Q,sigma);
            Print("\n#####\n");
            print := print - 1;
          fi;
        od;
      fi;
    fi;
  od;

end;

# -----------------------------------------------------

agrandar_matriz := function(max_row,max_col,matrix)

# Rutina que agranda la matriz matrix.

  local row;
  for row in [2,4..max_row] do
    agrandar_renglon(row,max_col,matrix);
  od;
```

```
end;

# --------------------------------------------------

funcs := [agrandar_matriz, agrandar_renglon, correspondencias,
       dibujar_correspondencia, calcula_contenciones,
       esta_contenida, relacion_contenciones,
       particiones_de_m_con_2_corazon_n, diferencia_de_nodos,
       diferencia_de_cuentas, encuentra_2_corazon,
       particion_triangular, encuentra_particion,
       encuentra_abaco, dibuja_abaco, particiones_2_regulares];
ProfileFunctions(funcs);

basename := "~/Documents/tesis/Ene19_1618.log";
LogTo(basename);
matrix := [];
agrandar_matriz(24,11,matrix);
SaveWorkspace(Concatenation(basename,".gap.bin"));
LogTo();

# --------------------------------------------------
```

Capítulo 6

Conclusiones

Como conclusión principal podemos decir que refutamos la unicidad de la matriz infinita de particiones 2-regulares conjeturada por Radha Kessar-Luis Valero Elizondo, en el artículo: “Stable partitions and Alperin’s weight conjecture for the symmetric groups in characteristic two”, donde se conjetura la existencia y unicidad de una matriz cuyas entradas son particiones que tengan las siguientes propiedades: (no mencionaremos la condición de pesos enunciada en el capítulo 3).

1. Que las particiones se vayan conteniendo por renglones.
2. Que su 2-corazón esté indexado por la columna.
3. Que sean particiones 2-regulares.
4. Que dentro de un mismo renglón la diferencia de los números de nodos de dos particiones es igual a la diferencia del número de nodos de sus respectivos dos corazones.

Cabe señalar que la existencia de dicha matriz no se ha podido refutar, en todo lo que trabajamos y todos los ejemplos que corrimos en la computadora, todo parece indicar que la matriz existe, sólo que no es única.

Como una segunda conclusión podemos decir que después de todo el trabajo de programación realizado en Gap, tenemos una biblioteca de rutinas para trabajar con particiones muy completa para poder ser utilizada por cualquiera que requiera trabajar con particiones.

Apéndice algebraico

En este apéndice veremos algunas definiciones y teoremas de Álgebra que se utilizan para la tesis, cabe señalar que ésta no es una tesis en Álgebra, sin embargo esta parte es necesaria para entender mejor de qué estamos hablando.

Sean G un grupo finito, p un número primo, y k un campo de descomposición para G de característica p. Todos nuestros módulos tendrán dimensión finita sobre k.

Definición: Un **peso** para G es una pareja (Q, S) donde Q es un p-subgrupo y S es un módulo simple para $k[N_G(Q)]$ que es proyectivo visto como un módulo para $k[N_G(Q)/Q]$.

Definición: Un λ-*tableau* es uno de los $n!$ arreglos de enteros obtenido reemplazando cada nodo en la partición λ por uno de los enteros $1, 2, \ldots, n$, sin repeticiones.

Definición: Si t es un *tableau*, su estabilizador por **renglones**, R_t, es el subgrupo de S_n que consta de los elementos que fijan todos los renglones de t por conjuntos.

.1. La conjetura de Alperin

Conjetura: En su artículo [Alperin, 1987], Alperin conjetura que si K es un campo algebraicamente cerrado, entonces el número de KG-módulos irreducibles (hasta isomorfismo) es igual al número de pesos (hasta conjugación) del KG-módulo regular.

Existe una versión más fuerte de esta conjetura, llamada la forma de bloques de la conjetura de Alperin. El siguiente texto está tomando de libro de [Benson, 1991]:

Si B es un bloque de KG con grupo de defecto D (no necesariamente abeliano), entonces el número de módulos simples en B es igual a la suma sobre todas las clases de conjugación de p-subgrupos P de G del número de módulos simples y proyectivos para $N_G(P)/P$, los cuales, cuando son vistos como módulos para $N_G(P)$ viven en un bloque b para el cual $b^G = B$. En otras palabras, el número de módulos simples en B, el cual puede ser pensado como un invariante global, es igual al número de correspondientes de Green simples de módulos en B, los cuales pueden ser calculados localmente.

álgebra de grupo KG [Ash, 2000] Sea K un campo cualquiera y G un grupo finito. Formamos el álgebra de grupo KG, la cual es un espacio vectorial sobre K y cuyos vectores base se corresponden con los elementos de G. En general, si $G = \{x_1, \ldots, x_m\}$, los elementos de KG son de la forma $\alpha_1 x_1 + \cdots + \alpha_m x_m$, donde los α_i pertenecen a K. La multiplicación en KG se define de la manera natural; hacemos $(\alpha x_i)(\beta x_j) = \alpha\beta x_i x_j$ y extendemos por linealidad. Entonces KG es un anillo (con identidad $1_K 1_G$) que es también un espacio vectorial sobre K, y $\alpha(xy) = (\alpha x)y = x(\alpha y)$, $\alpha \in K, x, y \in G$, así que KG es de hecho un álgebra sobre K.

KG**-módulo** Es un módulo cuyo anillo subyacente es el álgebra de grupo KG.

KG**-módulo regular** Es la manera en que nos referimos al álgebra de grupo KG cuando queremos hacer énfasis en sus propiedades como KG-módulo.

módulo irreducible Vea módulo simple.

módulo simple Es un módulo $M \neq 0$ cuyos únicos submódulos son 0 y M mismo.

peso [Alperin, 1987] Un peso para (K, G) es una pareja (Q, S) donde

1. Q es un p-subgrupo de G;
2. S es un $K\left[\frac{N_G(Q)}{Q}\right]$-módulo simple y proyectivo.

Bibliografía

[Alperin, 1987] J. L. Alperin. Weights for finite groups. In *The Arcata Conference on Representations of Finite Groups*, number 47 in Proceedings of symposia in pure mathematics, pages 369–379, Providence, R.I. American Mathematical Society (1987).

[Alperin and Fong, 1990] J. L. Alperin and P. Fong. Weights for symmetric and general linear groups. *Journal of Algebra*, 131:2–22 (1990).

[Ash, 2000] Robert B. Ash. *Abstract Algebra: The Basic Graduate Year*. Urbana, IL, U.S.A. (2000).

[Benson, 1991] David J. Benson. *Representations and cohomology I*, volume 30 of *Cambridge studies in advanced mathematics*. Cambridge University Press, Cambridge; New York (1991).

Índice alfabético