Vértices, Correspondientes de Green, Pesos y Conjetura de Alperin: Un enfoque Computacional

Adán Cortés Medina

Marzo 2006

[1] v.1.0β compilada el 23 de febrero de 2006

Índice general

Capítulo 1

Preliminares de la Teoría de Representaciones de Grupos

1.1. Estructuras Algebraicas

Las estructuras algebraicas básicas

Por completez del trabajo, comenzamos definiendo las estructuras algebraicas que vamos a necesitar.

1. Definiciones. Un **monoide** es una terna $(M, \cdot, 1)$, en la que M es un conjunto cualquiera; la función $\cdot : M \to M$, es asociativa (esto es, para todas $m_i \in M$ se tiene que $m_1 \cdot (m_2 \cdot m_3) = (m_1 \cdot m_2) \cdot m_3$, y por lo tanto no hay necesidad de usar paréntesis), por cierto, cuando estamos usando $\cdot$ para denotar la operación en el monomio, le llamamos multiplicación; finalmente, 1 es un elemento de M que tiene la siguiente propiedad: Para toda $m \in M$, $1 \cdot m = m \cdot 1 = m$. Para referirnos a esta propiedad, decimos que el 1 es (el) neutro bajo la multiplicación.

Un **grupo** es una clase particular de monoide, digamos $(G, \cdot, 1)$, en donde, además, pedimos que cada elemento de G tenga inverso, es decir: Para todo $g \in G$ existe en G un elemento, denotado g^{-1} tal que $g \cdot g^{-1} = g^{-1} \cdot g = 1$.

Cuando la operación del grupo es conmutativa, es decir $g_1 \cdot g_2 = g_2 \cdot g_1$ para toda $g_1, g_2 \in G$, entonces decimos que el grupo es **abeliano** y cambiamos de la notación $(G, \cdot, 1)$ a la notación $(G, +, 0)$.

Un **anillo** es una quinteta $(R, +, 0, \cdot, 1)$ en la que $(R, +, 0)$ es un grupo abeliano, $(R \backslash \{0\}, \cdot, 1)$ es un monoide y existen reglas de convivencia entre las dos estructuras, que mucha gente conoce como las **leyes distributivas**: $r_1 \cdot (r_2 + r_3) = (r_1 \cdot r_2) + (r_1 \cdot r_3)$; $(r_1 + r_2) \cdot r_3 = (r_1 \cdot r_3) + (r_2 \cdot r_3)$.

Un **campo** es una clase especial de anillos, digamos $(F, +, 0, \cdot, 1)$, en la que $(F \backslash \{0\}, \cdot, 1)$ no es sólo un monoide, sino un grupo.

Cuando es claro a partir del contexto cuales son las operaciones o los neutros de éstas, escribiremos solamente el conjunto o el conjunto y las operaciones. De esta manera un grupo $(G, \cdot, e)$ lo escribiremos la mayor parte del tiempo simplemente como G y, en algunos casos, como $(G, \cdot)$. Igualmente un anillo puede ser $(R, +, 0, \cdot, 1)$, $(R, +, \cdot)$ o simplemente R. Lo mismo aplica para las demás estructuras que definamos más adelante.

Acciones y G-Conjuntos

2. Definición. Sean R un anillo y X un grupo abeliano. Una función $\star : R \times X \to X$ se denomina una **multiplicación por escalares** si:

1. $1 \star x = x$ para toda $x \in X$;
2. $(r_1 + r_2) \star x = (r_1 \star x) + (r_2 \star x)$;
3. $(r_2 \cdot r_1) \star x = r_2 \star (r_1 \star x)$.

El concepto de acción recoge la esencia de lo que comúnmente conocemos como multiplicación por escalares. Comencemos por ver un par de ejemplos, y luego los iremos destilando hasta quedarnos con el caso más general.

3. Ejemplo. Sea $\mathbb{R}[x]$ el conjunto de todos los polinomios con coeficientes en $\mathbb{R}$. Esto es,

$$\mathbb{R}[x] = \{a_n x^n + \cdots + a_1 x + a_0 | a_0, \ldots, a_n \in \mathbb{R}\}.$$

Sabemos que los objetos en $\mathbb{R}[x]$, a los que solemos llamarles *polinomios* pueden ser sumados, restados y multiplicados entre sí, dando como resultado otro polinomio. También sabemos que los polinomios constantes 0 y 1 son, respectivamente, los neutros aditivo y multiplicativo. En otras palabras, los polinomios forman un anillo.

Pero, adicionalmente, los polinomios pueden ser multiplicados por escalares (ie. elementos de $\mathbb{R}$, que es el campo base), y cuando lo hacemos, éstos satisfacen las siguientes relaciones:

$$\begin{aligned} 1p(x) &= p(x) \\ (r+s)p(x) &= rp(x) + sp(x) \\ (rs)p(x) &= r(sp(x)) \end{aligned}$$

para todas las r y $s \in \mathbb{R}$ y para todo $p(x) \in \mathbb{R}[x]$.

4. Ejemplo. Denotemos por $\mathbb{Z}^3$ al conjunto $\{(x, y, z) | x, y, x \in \mathbb{Z}\}$. A los elementos de $\mathbb{Z}^3$ se les suele llamar *vectores* y podemos pensarlos como los puntos del espacio que tienen sus tres coordenadas enteras.

A diferencia de los polinomios, los vectores en $\mathbb{Z}^3$ no se pueden multiplicar. Sin embargo, podemos sumarlos y restarlos, y, cuando lo hacemos, obtenemos otro vector en $\mathbb{Z}^3$. Finalmente, con el vector $(0, 0, 0)$ como neutro aditivo, tenemos una estructura de grupo.

Los vectores (en general) también se pueden multiplicar por escalares, en este caso los escalares serían los elementos de $\mathbb{Z}$, y satisfarán las siguientes propiedades:

$$\begin{aligned} 1(x, y, z) &= (x, y, z) \\ (r+s)(x, y, z) &= r(x, y, z) + s(x, y, z) \\ (rs)(x, y, z) &= r(s(x, y, z)) \end{aligned}$$

para todas las r y $s \in \mathbb{Z}$ y para todo vector $(x, y, z) \in \mathbb{Z}^3$.

Como se observa de los ejemplos anteriores, hay tres cosas que son básicas para tener una multiplicación por escalares: La primera es, obviamente, los escalares; la segunda un conjunto de objetos que vayan a ser multiplicados por escalares; y la tercera las reglas de como se comportan los escalares cuando hay más de uno de ellos. También hay

que notar un gran faltante: No necesitamos ni la estructura de anillo de los polinomios nos bastó que éstos forman un grupo abeliano.

Si comenzamos con un conjunto X que no tenga estructura algebraica alguna, podemos aún definir una especie de multiplicación por escalares, la cual se conoce como acción.

5. Definiciones. Sean $(G, \cdot, e)$ un grupo y X un conjunto no vacío. Una función $\star : G \times X \to X$ se denomina una **acción** de G en X si:

1. $e \star x = x$;
2. $(g_2 \cdot g_1) \star x = g_2 \star (g_1 \star x)$.

para toda $x \in X$ y para todas g_1 y $g_2 \in G$.

Cuando es claro a cual acción nos referimos, a veces, simplemente decimos que X es un **G-conjunto**

Note la costumbre de escribir $g \star x$ en vez de $\star(g, x)$. Esto nos facilita la escritura, a la vez que nos hace sentir que estamos multiplicando.

6. Ejemplos. Los ejemplos clásicos de acciones son los siguientes, en todos los casos G es un grupo:

1. *Multiplicación por la izquierda:* Cuando $X = G$, podemos definir la acción de G en si mismo por $g \star x = gx$.
2. *Conjugación en Elementos:* Cuando $X = G$, otra manera de definir la acción es mediante $g \star x = gxg^{-1}$;

   Es una acción porque, como $(gh)^{-1} = h^{-1}g^{-1}$, tenemos que $(gh)\star x = (gh)x(gh)^{-1} = g(hxh^{-1})g^{-1} = g \star (h \star x)$.
3. *Conjugación en Subgrupos:* Cuando X es el conjunto de todos los subgrupos de G (es decir, $X = \{H | H \leq G\}$) se puede definir $g \star H = gHg^{-1}$;
4. *Multiplicación en Clases Laterales Izquierdas* Si H es un subgrupo de G, entonces podemos definir una acción de G en las clases laterales izquierdas de H: Tomamos $X = \{xH | x \in G\}$ y definimos $g \star xH = (gx)H$;
5. *Conjugación en Subconjuntos* Cuando X es $\mathscr{P}(G)$ el conjunto de todos los subconjuntos de G se puede definir $g \star S = gSg^{-1}$;
6. *Multiplicación en Subconjuntos* Si S es un subconjunto de G, entonces podemos definir $g \star S = gS$;
7. *La Trivial* Para un conjunto arbitrario X, $g \star x = x$.

7. Definición. Si X es un G-conjunto, definimos los **puntos fijos** de X bajo la acción de $\star$ como el conjunto $\{x \in X \mid g \star x = x \quad \forall g \in G\}$, y lo denotamos X^G.

Más estructuras algebraicas

Ahora definiremos las tres estructuras que nos faltan.

8. Definiciones. Un **espacio vectorial** V sobre el campo K es una estructura formada por 3 elementos: Un campo $(K, +, 0, \cdot, 1)$; un grupo abeliano $(V, \oplus, \vec{0})$; y una función de multiplicación por escalares $\star : K \times V \to V$.

Al campo K muchas veces se le nombra el **campo de escalares**; a los elementos de V se les conoce como **vectores**; a la operación $\oplus$ de V se le llama **suma vectorial** y al neutro de la suma vectorial se le conoce como el **vector cero** o como el **origen**.

Como la multiplicación por escalares $\star$ la definimos sobre un grupo abeliano (que no sobre un conjunto), pedimos que adicionalmente $\star$ se distribuya sobre la suma vectorial.

Formalmente, pedimos que la función $\star$ satisfaga:

1. $1 \star v = v$;
2. $(k_1 \cdot k_2) \star v = k_1 \star (k_2 \star v)$, esto es, $\star$ es una acción.
3. $(k_1 + k_2) \star v = (k_1 \star v) \oplus (k_2 \star v)$;
4. $k \star (v_1 \oplus v_2) = (k \star v_1) \oplus (k \star v_2)$.

9. Ejemplos.

1. Si K es un campo, entonces K^n, el conjunto de todas las n-adas $(k_1, k_2, \ldots, k_n)$, con la suma coordenada a coordenada y la multiplicación definida por $k \star (k_1, k_2, \ldots, k_n) = (k \cdot k_1, k \cdot k_2, \ldots, k \cdot k_n)$ forma un espacio vectorial sobre K;
2. Las matrices de tamaño $m \times n$ con coeficientes en un campo K forman un espacio vectorial sobre K.

10. Observación. Recordemos que un campo se define como un anillo conmutativo con división. Si volvemos a ver la definición de espacio vectorial, notaremos que en ningún momento ocupamos que el producto $\cdot$ fuera conmutativo, y la definición tampoco depende de la existencia de los inversos multiplicativos del campo.

Por otro lado, no hay ninguna razón por la cual la multiplicación por escalares tenga que ser por la izquierda (ie. $\star : K \times V \to V$), en realidad podríamos pedir que fuera por la derecha (ie. $\star : V \times K \to V$).

Entonces, ¿por qué no buscar una generalización natural del concepto de espacio vectorial?. El concepto de **módulo** es dicha generalización, de hecho, se definen módulos izquierdos y derechos, en los módulos izquierdos la multiplicación por escalares se hace por la izquierda y en los módulos derechos se hace por la derecha, pero casi siempre que uno habla de módulos está pensando en módulos izquierdos, así que definiremos solamente módulo izquierdo y cuando digamos módulo se debe entender módulo izquierdo.

11. Definiciones. Un **módulo (izquierdo)** es una estructura compuesta por un anillo $(R, +, 0, \cdot, 1)$ (conocido como el **anillo subyacente**), un grupo abeliano $(M, \oplus, \vec{0})$ y una acción $\star : R \times M \to M$; que satisfacen las siguientes reglas, para toda $m_i \in M$ y toda $r_j \in R$:

1. $1 \star m = m$;
2. $(r_1 \cdot r_2) \star m = r_1 \star (r_2 \star m)$, es decir, $\star$ es un acción;
3. $(r_1 + r_2) \star m = (r_1 \star m) \oplus (r_2 \star m)$;
4. $r \star (m_1 \oplus m_2) = (r \star m_1) \oplus (r \star m_2)$.

Con la finalidad de fijar ideas diremos que un **módulo** es un conjunto de objetos llamados **vectores** que se pueden sumar, restar y multiplicar por **escalares** (elementos del anillo subyacente) y que obedecen reglas para la convivencia de la suma vectorial con la multiplicación por escalares.

Para resaltar al anillo subyacente, es común decir que el módulo M es un R**-módulo**, o que es un módulo sobre R.

12. Observación. Notemos que podemos, sin generar confusiones, llamar $+$ tanto a la suma vectorial $\oplus$ como a la suma $+$ del anillo; así como omitir los operadores $\cdot$ y $\star$, ya que el conocer a que estructura algebraica pertenece cada elemento nos permite saber a que suma o a que producto nos referimos. Sean $m_1, m_2 \in M$ y $r_1, r_2 \in R$. Veamos todos los casos:

La expresión	se refiere a
$m_1 + m_2$	$m_1 \oplus m_2$
$r_1 + r_2$	$r_1 + r_2$
$r_1 m_2$	$r_1 \star m_2$
$r_1 r_2$	$r_1 \cdot r_2$

En los demás casos la operación no está definida.

En adelante usaremos $+$ para denotar tanto la suma vectorial $\oplus$ como la suma de escalares $+$; y cualquier yuxtaposición de caracteres significará la multiplicación $\cdot$ o $\star$, dependiendo del contexto.

Además, como para todo $m \in M$ se tiene que $0 \star m = \vec{0}$, escribiremos 0 tanto para el cero del anillo como para el vector cero, $\vec{0}$.

13. Ejemplos.

1. Todo espacio vectorial sobre un campo K es un K-módulo;
2. Todo anillo R es un módulo sobre sí mismo;
3. Si R es un anillo, entonces R^n el conjunto de todas las n-adas $(r_1, r_2, \ldots, r_n)$, donde cada $r_i \in R$ es un módulo sobre R;
4. Las matrices de tamaño $m \times n$ con coeficientes en un anillo R son un R-módulo;
5. Todo grupo abeliano G es un módulo sobre $\mathbb{Z}$ definiendo la multiplicación $ng = g + \cdots + g$, donde el lado derecho tiene n sumandos;
6. Sea I un ideal izquierdo de R. Entonces I es un módulo izquierdo sobre R.

La siguiente estructura que nos interesa es un **álgebra**. Podemos pensar en un álgebra como un módulo donde también se permite la multiplicación (vectorial).

14. Definición. Un **álgebra** (asociativa) es una estructura formada por dos anillos, $(R, +, 0, \cdot, 1)$ y $(A, \oplus, \vec{0}, \otimes, \vec{1})$; una multiplicación por escalares $\star : R \times A \to A$, tales que satisfacen, para toda $a_i \in A$ y toda $r_i \in R$:

1. $1 \star a = a$;
2. $(r_1 \cdot r_2) \star a = r_1 \star (r_2 \star a)$, es decir, $\star$ es un acción;
3. $(r_1 + r_2) \star a = (r_1 \star a) \oplus (r_2 \star a)$;
4. $r \star (a_1 \oplus a_2) = (r \star a_1) \oplus (r \star a_2)$;

5. $r \star (a_1 \otimes a_2) = (r \star a_1) \otimes a_2 = a_1 \otimes (r \star a_2)$

Note que, en este caso, la multiplicación por escalares se definió sobre un anillo (y no simplemente sobre un conjunto), así que tuvimos que agregar a las propiedades de $\star$ un par de reglas para cuando la acción se aplica a la suma o al producto de vectores.

15. Ejemplos.

1. Todo anillo conmutativo R es un álgebra sobre sí mismo;
2. Un anillo arbitrario R es siempre una $\mathbb{Z}$-álgebra.
3. Las matrices cuadradas (i.e. de tamaño $n \times n$) con coeficientes en un anillo R forman una R-álgebra.
4. Si R es un anillo conmutativo, el anillo de polinomios sobre R, $R[x]$ es una R-álgebra, al igual que el anillo de series formales $R[[x]]$.
5. Si E/F es una extensión de campos, entonces E es un álgebra sobre F.

16. Ejemplo. Sean G un grupo finito, digamos $G = \{g_1 = 1, g_2, \dots g_n\}$; y K un campo. Sea A el espacio vectorial sobre K que tiene como base a G (esto es, el vector $(k_1, k_2, \dots, k_n) \in A$ representa la suma $\sum_{i=1}^{n} k_i g_i$), y definimos la multiplicación entre dos vectores como la multiplicación de las combinaciones lineales que representan (es decir,

$$\sum_{i=1}^{n} k_i g_i \sum_{j=1}^{n} l_j g_j = \sum_{i,j} (k_i l_j) g_{ij},$$

donde g_{ij} es el producto en G de g_i con g_j).

Note que el producto en A es asociativo, pues los productos en el campo K y en el grupo G lo son.

También es claro que, si $k \in K$, entonces

$$k(\sum_{i=1}^{n} k_i g_i \sum_{j=1}^{n} l_j g_j) = (k \sum_{i=1}^{n} k_i g_i)(\sum_{j=1}^{n} l_j g_j) = (\sum_{i=1}^{n} k_i g_i)(k \sum_{j=1}^{n} l_j g_j)$$

pues el producto en el campo K es conmutativo.

Entonces A es un álgebra.

17. Definición. Sean G un grupo finito y K un campo. El **álgebra de grupo** KG es el álgebra obtenida en el ejemplo anterior.

Buena parte de este trabajo se basará en el estudio de las álgebras de grupo.

1.2. KG-módulos

Lo que se presenta en seguida son sólo las definiciones y una lista de los resultados necesarios para este trabajo. El lector encontrará una exposición más completa y con todas las demostraciones en [Curtis and Reiner, 1990]. Si sólo desea una introducción, el libro de [Ash, 2000] es también una buena opción. El libro de [Alperin, 1986] es por supuesto una referencia obligada.

Hemos visto que KG es un álgebra, así que, en particular es un anillo. Ahora nos interesa estudiar los módulos sobre el anillo KG.

18. Observación. Otra manera de definir a los KG-módulos es la siguiente: Decimos que M es un KG-módulo si es al mismo tiempo un K-espacio vectorial y un G-conjunto que satisface las siguientes reglas de convivencia entre las dos estructuras:

1. $(k_1 + k_2)m = (k_1 m) + (k_2 m)$;
2. $k(m_1 + m_2) = (km_1) + (km_2)$.

Esto tiene que ver con que un típico elemento de KG es una suma finita de la forma $\sum_i k_i g_i$. Y, si $m \in M$, entonces

$$\left(\sum_i k_i g_i\right) \star m = \sum_i (k_i g_i)m = \sum_i k_i (g_i m) = \sum_i k_i m'$$

por lo tanto, basta conocer la acción de G en M. Más aún, si $\{\mathfrak{g}_1, \ldots, \mathfrak{g}_n\}$ es un conjunto de generadores de G, basta conocer la acción de éstos para que exista una única extensión a todo el grupo y luego a todo el KG-módulo.

19. Nota. Todos los KG-módulos en este trabajo son de dimensión finita (al ser vistos como espacios vectoriales).

Representación de KG-módulos

Hemos visto que si M es un KG-módulo, entonces puede ser visto como un K-espacio vectorial que es también un G-conjunto.

20. Observación. Para cada $g \in G$, denotemos por $\rho_g : M \to M$ a la transformación lineal determinada por la acción, esto es $\rho_g(m) = gm$. Es claro que ρ_g es una transformación lineal si notamos que g se puede escribir como un vector que todas sus entradas excepto una, son cero y vemos la definición del producto de dos elementos en M.

Si ahora definimos $\rho : G \to GL(M)$ como la función que a cada $g \in G$ lo manda en la transformación lineal ρ_g del párrafo anterior, tenemos que ρ es un homomorfismo pues $\rho(g_1 g_2)(m) = \rho_{g_1 g_2}(m) = (g_1 g_2)m = g_1(g_2 m) = \rho_{g_1}(\rho_{g_2}(m)) = (\rho_{g_1} \circ \rho_{g_2})(m) = (\rho(g_1) \circ \rho(g_2))(m)$. Además cada $\rho(g)$ es invertible ya que $\rho(g)\rho(g^{-1}) = \rho(1_G)$ que es la identidad en M.

21. Definiciones. El homomorfismo ρ de la observación anterior se conoce como la **representación** de G en el KG-módulo M. Si queremos, podemos remplazar a $GL(M)$ por el grupo de todas las matrices no singulares de n por n con entradas en K. En ese caso ρ es llamada una **representación matricial**.

22. Observación. El proceso anterior puede ser revertido. Dados un K-espacio vectorial V y un homomorfismo ρ de G en las transformaciones lineales invertibles $GL(V)$, podemos definir una acción lineal de G en V por $gv = \rho(g)(v)$, y por tanto hacer de V un KG-módulo. Así las representaciones se pueden identificar con los KG-módulos.

El KG-módulo regular

23. Definición. De la misma manera que un anillo R puede ser visto como R-módulo, el álgebra de grupo KG puede ser vista como un KG-módulo, éste se conoce como el KG-módulo **regular**.

24. Ejemplo. El KG-módulo regular $\mathbb{F}_2\mathbb{Z}_2$. Si denotamos a los elementos de $\mathbb{Z}_2$ por e y x, los elementos de $\mathbb{F}_2\mathbb{Z}_2$ serán $\vec{0} = 0e + 0x, e = 1e + 0x, x = 0e + 1x$ y $e + x = 1e + 1x$.

La suma vectorial esta dada por:

$+$	$\vec{0}$	e	x	$e+x$
$\vec{0}$	$\vec{0}$	e	x	$e+x$
e	e	$\vec{0}$	$e+x$	x
x	x	$e+x$	$\vec{0}$	e
$e+x$	$e+x$	x	e	$\vec{0}$

y la multiplicación por elementos del anillo KG por:

$\cdot$	$\vec{0}$	e	x	$e+x$
$\vec{0}$	$\vec{0}$	$\vec{0}$	$\vec{0}$	$\vec{0}$
e	$\vec{0}$	e	x	$e+x$
x	$\vec{0}$	x	e	$e+x$
$e+x$	$\vec{0}$	$e+x$	$e+x$	$\vec{0}$

25. Contraejemplo. En el caso de los espacios vectoriales tenemos el siguiente resultado: Sea V un K-espacio vectorial. Si $0 \neq k \in K$ y $\vec{0} \neq v \in V$, entonces $kv \neq \vec{0}$. Como podemos ver en la tabla de multiplicación por escalares del ejemplo anterior, el resultado no se cumple para KG-módulos: $0 \neq e+x$ es un escalar, y $\vec{0} \neq e+x$ es también un vector en $\mathbb{F}_2\mathbb{Z}_2$, sin embargo, $(e+x)(e+x) = \vec{0}$.

Homomorfismos de KG-módulos

26. Definiciones. Sean M y N dos KG-módulos. Un **homomorfismo** de KG-módulos de M a N es una aplicación $\varphi : M \to N$ tal que, para toda $x, y \in M$ y toda $r, s \in KG$, se cumple que:

$$\varphi(rx + sy) = r\varphi(x) + s\varphi(y)$$

El **kernel** de φ es el conjunto $\{x \in M | \varphi(x) = 0\}$ y la **imagen** es el conjunto $\{\varphi(x) | x \in M\}$. Usualmente los denotamos $\ker(\varphi)$ e $\mathrm{Im}(\varphi)$, respectivamente.

27. Definiciones. Decimos que el homomorfismo de KG-módulos $\varphi : M \to N$ es: un **isomorfismo** si, como función, es biyectiva; un **monomorfismo** si, como función, es inyectiva; un **epimorfismo** si, como función, es suprayectiva;

28. Definiciones. Si φ es un homomorfismo del KG-módulo M en sí mismo (ie. $\varphi : M \to M$), entonces le llamamos **endomorfismo**. Un **automorfismo** es un endomorfismo que además es isomorfismo.

Al conjunto de todos los endomorfismos del KG-módulo M (al cual, por cierto, se le puede dar estructura de KG-módulo) lo denotamos $\mathbf{End}_{KG}(M)$. Finalmente, $\mathbf{Aut}_{KG}(M)$ denota a los automorfimos del KG-módulo M.

KG-submódulos

29. Definición. Si N es un subconjunto no vacío del KG-módulo M, decimos que N es un KG**-submódulo** de M (y escribimos $N \leq M$) si para cada $x, y \in N$ y cada $r, s \in KG$ tenemos que $rx + sy \in N$.

En particular, N es un K-subespacio vectorial y un G-subconjunto de M.

30. Ejemplo. Todo KG-módulo M tiene al menos dos submódulos:

1. El primero de ellos es $\{\vec{0}\}$. Es claro que si $\vec{0} \in M$, entonces $\{\vec{0}\}$ es submódulo. Veamos que $\vec{0} \in M$: $0 = 0e \in KG$; sea $m \in M$. Luego, $0m = \vec{0} \in M$.

2. Claramente $M \leq M$.

El KG-módulo trivial

31. Definición. Sea KG un álgebra de grupo. El conjunto $M = \{\sum_{g \in G} kg : k \in K\}$ se puede ver como un submódulo del KG-módulo regular. A este submódulo o a cualquiera isomorfo a él se les conoce como el KG-módulo **trivial**.

Note que la aplicación $\varphi : K \to M$ que manda a k en $\sum_{g \in G} kg$ es un isomorfismo. Es por esto que al KG-módulo trivial se le denota usualmente como K.

KG-módulos simples e inescindibles

32. Definición. Decimos que el KG-módulo $M \neq 0$ es **simple** (o **irreducible**) si sus únicos submódulos son $\{\vec{0}\}$ y M mismo.

33. Ejemplo. Si observamos las tablas de suma y multiplicación por escalares del KG-módulo regular $\mathbb{F}_2\mathbb{Z}_2$ vemos que tiene un submódulo no trivial, a saber $\{\vec{0}, e + x\}$, por lo que el módulo regular $\mathbb{F}_2\mathbb{Z}_2$ no es simple. Aquí están sus tablas de suma y multiplicación por escalares:

$+$	$\vec{0}$	$e+x$
$\vec{0}$	$\vec{0}$	$e+x$
$e+x$	$e+x$	$\vec{0}$

$\cdot$	$\vec{0}$	$e+x$
$\vec{0}$	$\vec{0}$	$\vec{0}$
e	$\vec{0}$	$e+x$
x	$\vec{0}$	$e+x$
$e+x$	$\vec{0}$	$\vec{0}$

34. Definiciones. Una **serie** de longitud n para un módulo M es una sucesión de la forma $M = M_0 \geq M_1 \geq \cdots \geq M_n = 0$. La serie es llamada **serie de composición** si cada uno de los módulos cocientes M_i/M_{i+1} es simple.

Por convención, el módulo $\{0\}$ tiene como serie de composición a $\{0\}$.

Decimos que dos series son **equivalentes** si tienen la misma longitud y los mismos módulos cociente hasta isomorfismo y reordenamiento.

35. Observación. Una serie de descomposición no tiene reordenamientos propios.

36. Proposición. Todo KG-módulo M de dimensión finita, tiene una serie de descomposición.

Demostración: Por inducción sobre la dimensión de M. Si $\dim(M) = 1$, entonces M es, claramente, simple. Su serie de composición es $M \geq 0$. Supongamos que todo módulo de dimensión menor que n tiene serie de composición y tomemos un módulo M de dimensión n. Si M es simple, su serie de composición es $M \geq 0$; si M no es simple, debe tener un submódulo maximal M_1, luego M/M_1 es simple y M_1 tiene una serie de composición $M_2 \geq \cdots \geq M_k$. Entonces $M \geq M_1 \geq M_2 \geq \cdots \geq M_k$ es serie de composición para M. ∎

37. Definiciones. Sea M_i, con $i \in I$, una familia de KG-módulos. Su **suma directa externa**, la cual denotamos $\bigoplus_i M_i$,consiste en todas las familias $(a_i, i \in I)$, con $a_i \in$

M_i, tales que solamente una cantidad finita de las a_i es diferente de cero. La suma se define por $(a_i) + (b_i) = (a_i + b_i)$ y la multiplicación por escalares por $r(a_i) = (ra_i)$.

Decimos que el KG-módulo M es la **suma directa interna** de los submódulos M_i si cada $x \in M$, $x \neq 0$, puede ser expresada de manera única como $x_{i_1} + \cdots + x_{i_n}$ donde los i_k se asumen distintos para $k = 1, \ldots, n$, $x_{i_k} \neq 0$ pertenece a M_{i_k}, y tanto el entero positivo n como los elementos x_{i_k} dependen de x. La denotamos igual que la suma directa externa $\bigoplus_i M_i$. Y a cada uno de los M_i se les llama **sumando directo interno** y se denota $M_i \mid M$

En particular, si $\bigoplus_i M_i$ es una suma directa interna para el KG-módulo M, también es una suma directa de espacios vectoriales.

38. Definición. Sea M un KG-módulo. Decimos que M es **semisimple** si M es suma directa de submódulos simples.

En particular, todo simple es semisimple.

El siguiente teorema nos servirá para no tener que pedirle a GAP que haga ejemplos donde p no divida al $|G|$. Vea la demostración en el libro de [Ash, 2000] sec. 9.6.

39. Teorema. (Maschke) Sean G un grupo finito G y p la característica del campo K. Entonces p no divide a $|G|$ si y sólo si el KG-módulo regular es semisimple.

Vea la demostración en [Ash, 2000] capítulo 4, p. 19.

40. Definición. Decimos que el KG-módulo M es **inescindible** si no se puede escribir como suma directa interna de submódulos propios.

41. Ejemplos.

1. Todo KG-módulo simple es inescindible;
2. El KG-módulo regular $\mathbb{F}_2\mathbb{Z}_2$ es inescindible.

42. Observación. Las definiciones de simple e inescindible no son equivalentes, $\mathbb{F}_2\mathbb{Z}_2$ es inescindible, pero no es simple.

43. Proposición. Todo KG-módulo de dimensión finita se puede descomponer como suma de submódulos inescindibles.

44. Definición. Un anillo R se dice **local** si cumple cualquiera de las siguientes condiciones equivalentes (Para una demostración vea [Curtis and Reiner, 1990] §5C):

1. R tiene un único ideal maximal izquierdo
2. Denotemos por U al conjunto de unidades en R. El conjunto $R - U$ de las no unidades de R es un ideal izquierdo.

45. Proposición. Sea M un KG-módulo diferente de cero de dimensión finita. Entonces M es inescindible si y sólo si el anillo $\mathrm{End}_{KG}(M)$ es local

46. Teorema. (Krull-Smidth-Azumaya) Si M es un KG-módulo que se puede escribir como dos sumas distintas de KG-módulos inescindibles, digamos

$$M = \bigoplus_{i=1}^{r} M_i = \bigoplus_{j=1}^{s} N_j \tag{1.1}$$

entonces $r = s$ y, después de una renumeración, $M_i \cong N_i$.

Demostración: Por inducción sobre el valor de r de la ecuación 1.1.

Primero notemos que si $r = 1$, entonces M es inescindible, pues $M = M_1$ y, por lo tanto $s = 1$ y $M_1 = N_1$.

Supongamos ahora que $r \geq 2$. Sean $\mu_i : M \to M_i$ y $\nu_j : M \to N_j$ las proyecciones asociadas a la descomposición de la ecuación 1.1.

Como las ν_j son proyecciones, $\sum \nu_j = \mathrm{id}_M$ y, componiendo con μ_1 tenemos:

$$\mu_1 \sum \nu_j = \sum \mu_1 \nu_j = \text{ mapeo identidad en } M_1,$$

y donde cada aplicación $\mu_1\nu_j$ está restringida a M_1. Sin embargo, como M_1 es inescindible (usando 45) tenemos que el anillo de endomorfismos $\mathrm{End}_A(M_1)$ es local, esto es, la suma de dos elementos no-unidad en este anillo es a su vez no-unidad. La ecuación de arriba entonces implica que alguna $\mu_1\nu_j$ es un automorfismo de M_1. Renumerando los N_j si fuera necesario, podemos asumir que $\mu_1\nu_1 = \varphi \in Aut_A(M_1)$. Pero entonces en el diagrama

$$M_1 \underset{\varphi^{-1}\mu_1}{\overset{\nu_1}{\rightleftarrows}} N_1$$

tenemos que $(\varphi^{-1}\mu_1)\nu_1 = \mathrm{id}_{M_1}$, de donde $\nu_1 M_1$ es un sumando directo de N_1. Como tanto M_1 como N_1 son inescindibles, se sigue que ν_1 es un isomorfismo $M_1 \cong N_1$, y que $\varphi^{-1}\mu_1 : N_1 \to M_1$ es su inversa.

$$M' = N_1 \oplus M_2 \oplus \cdots \oplus M_r.$$

(Note que la suma es directa, puesto que si

$$n_1 + m_2 + \cdots + m_r = 0, n_1 \in N_1, m_j \in M_j,$$

entonces aplicando mu_1 vemos que $\mu_1(n_1) = 0$, de donde $n_1 = 0$. Pero entonces cada $m_j = 0$, $j = 2, \ldots, r$.) En seguida, para $x \in N_1$ tenemos

$$\mu_1(x) = x - \mu_2(x) - \cdots - \mu_r(x) \in M',$$

y así $M_1 = \mu_1(N_1) \subset M'$. Por lo tanto $M' = M$, así que tenemos

$$M = M_1 \oplus M_2 \oplus \cdots \oplus M_r = N_1 \oplus M_2 \oplus \cdots \oplus M_r.$$

Ahora hacemos $\rho = \nu_1\mu_1 + \mu_2 + \cdots + \mu_r$; claramente ρ define un A-automorfismo de M que lleva a M_1 sobre N_1, y cada M_j sobre sí mismo para $j = 2, \ldots, r$. Así ρ induce un A-automorfismo $M/M_1 \cong M/N_1$; pero $M/N_1 = \oplus_{j=2}^{s} N_j$ así que tenemos que $\sum_{i=2}^{r} M_i \cong \sum_{j=2}^{2} N_j$. Ahora usamos la hipótesis de inducción para concluir que $\{M_i : 2 \leq i \leq r\}$ son, hasta isomorfismo, un reacomodo de los $\{N_j : 2 \leq j \leq s\}$. Esto completa la prueba. ∎

Generadores y Bases de KG-módulos

47. Definiciones. Sea S un conjunto de vectores del KG-módulo M. Decimos que S es un conjunto de **generadores** si toda $x \in M$ se puede escribir como una combinación lineal finita de elementos de S con coeficientes en KG.

Decimos que el conjunto S es linealmente independiente sobre KG si cada vez que una combinación lineal $\sum_{i=1}^{k} r_i x_i$ de elementos x_i de S con coeficientes r_i en KG es igual a $\vec{0}$, se tiene que todos los coeficientes son 0. (ie. $r_i = 0$ para $i = 1, \ldots, k$).

48. Definición. Una **base** es un conjunto de generadores linealmente independiente.

49. Observación. A diferencia de los espacios vectoriales, no todas las posibles bases para un KG-módulo M tienen la misma cardinalidad, a menos que KG sea conmutativo.

50. Contraejemplo. No todos los módulos tienen base: Los racionales $\mathbb{Q}$ como $\mathbb{Z}$-módulo son un módulo que no tiene base: Es claro que no existe un racional q_0 que genere a todo $\mathbb{Q}$, porque entonces $\mathbb{Q}$ sería igual a $q_0\mathbb{Z}$; por otro lado, cualquier conjunto de dos o más racionales es linealmente dependiente (y por lo tanto no pueden formar una base): Sean $p_0/q_0, p_1/q_1, \ldots, p_r/q_r$ racionales, basta encontrar α_i $(i = 0, \ldots, r)$ no todas igual a cero tales que satisfagan $\frac{p_0}{q_0}\alpha_0 + \frac{p_1}{q_1}\alpha_1 + \ldots \frac{p_r}{q_r}\alpha_r = 0$. Tomemos $\alpha_0 = p_1 q_0, \alpha_1 = -p_0 q_1$, y las demás igual a cero. Esto satisface la ecuación anterior.

51. Contraejemplo. Un KG-módulo que no tiene base: Sea M submódulo de $\mathbb{F}_2\mathbb{Z}_2$ del ejemplo de la sección anterior, esto es $M = \{\vec{0}, e+x\}$. De todos los subconjuntos de los elementos de M, los dos que son conjuntos de generadores son $\{e+x\}$ y $\{\vec{0}, e+x\}$. Pero ninguno de ellos es un conjunto de generadores linealmente independientes, pues en el primer caso, tenemos que $(e+x)\cdot(e+x) = \vec{0}$, y en el segundo caso $e\cdot\vec{0}+(e+x)\cdot(e+x) = \vec{0}$.

KG-módulos Libres y Proyectivos

En la sección anterior acabamos de ver que no todos los módulos tienen base. Aquellos que sí la tienen (o que cumplen cualquiera de las condiciones equivalentes del siguiente teorema) se conocen como libres.

52. Teorema. Sea M un KG-módulo. Las siguientes afirmaciones son equivalentes:

1. Existe una base para M;
2. existe una $n \in \mathbb{N}$ tal que M es isomorfo a $\bigoplus_{i=1}^{n} KG$;
3. existe $B = \{b_1, b_2, \ldots, b_n\} \subset M$ tal que para todo KG-módulo N y para todo n_i (con $i = 1, \ldots, n$, y $n_i \in N$) existe un único homomorfismo de KG-módulos $\varphi : M \to N$ tal que $\varphi(x_i) = n_i$.
4. existe X, subespacio de M, tal que cualquier transformación lineal φ de X en cualquier KG-módulo N se extiende de manera única a un homomorfismo de KG-módulos $\Phi : M \to N$.

53. Definiciones. Un KG-módulo M que cumple con cualquiera de las condiciones equivalentes del teorema anterior se denomina **libre**. En este caso, se define el **rango** del KG-módulo libre M como el natural n de la segunda equivalencia del teorema anterior, el cual es único.

54. Ejemplo. KG es libre para cualquier campo K y cualquier grupo G.

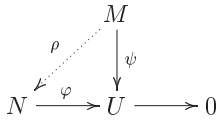
55. Observación. Si $M \neq 0$ es un KG-módulo tal que su cardinalidad es menor que la de KG, entonces M no puede ser libre.

56. Contraejemplos.

1. Si G es no trivial, entonces el KG-módulo trivial K no puede ser libre.
2. Sea M el $\mathbb{F}_2\mathbb{Z}_2$-módulo $\{\vec{0}, e+x\}$. Como $\{\vec{0}, e+x\} \subsetneq \mathbb{F}_2\mathbb{Z}_2$, M no es libre. De hecho, ningún sumando directo propio del KG-módulo regular es libre.
3. $\mathbb{Q}$ visto como $\mathbb{Z}$-módulo tampoco es libre.

57. Teorema. Sea M un KG-módulo. Las siguientes afirmaciones son equivalentes:

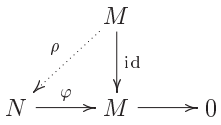
1. M es un sumando directo de un KG-módulo libre;
2. si φ es un epimorfismo del KG-módulo N sobre M, entonces, el kernel de φ es un sumando directo de N;
3. si φ es un epimorfismo del KG-módulo N sobre U, y ψ es un homomorfismo de M a U, entonces existe un homomorfismo $\rho : M \to N$ tal que $\varphi\rho = \psi$.



Para una prueba, véase [Alperin, 1986] pp. 29-31.

58. Definición. Decimos que un KG-módulo M es **proyectivo** si cumple cualquiera de las condiciones equivalentes del teorema anterior.

59. Observación. Si M es un KG-módulo proyectivo y $\varphi : N \to M$ es un epimomorfismo de KG-módulos, del diagrama en la definición de proyectivo tenemos:



y por lo tanto existe ρ tal que $\varphi\rho = \mathrm{id}_M$.

60. Observación. Se sigue de la definición que todo KG-módulo libre es proyectivo.

61. Contraejemplo. No todo módulo proyectivo es libre.

$\mathbb{F}_2 S_3$ es un KG-módulo regular y, por lo tanto, libre. Sin embargo no es inescidible, se parte como $\mathbb{F}_2 \oplus M \oplus M$ donde M, visto como $\mathbb{F}_2$-espacio vectorial, tiene dimensión 1.

Entonces, M es proyectivo pero no es libre.

$\mathbf{End}_K(M)$ como KG-módulo

62. Definición. Sea M un KG-módulo. Denotemos por $\mathbf{End}_K(M)$ al anillo de endomorfismos de M visto solamente como espacio vectorial, esto es, al conjunto $\{f : M \to M | f \text{ es } K \text{ lineal.}\}$

Para $g \in G$ y $f \in \mathrm{End}_K(M)$, definimos $\star : G \times \mathrm{End}_K(M) \to \mathrm{End}_K(M)$ como $(g \star f)(m) = gf(g^{-1}m)$, para todas las $g \in G$, $f \in \mathrm{End}_K(M)$ y $m \in M$.

63. Proposición. La aplicación $\star$ es una acción. Por tanto, $\mathrm{End}_K(M)$ tiene estructura de KG-módulo.

Demostración:

1. $1 \star f = f$:
$$(1 \star f)(m) = 1f(1m) = f(m).$$
2. $(g_1 g_2) \star f = g_1 \star (g_2 \star f)$:
$$(g_1 \star (g_2 \star f))(m) = g_1(g_2 \star f)(g_1^{-1}m) = g_1 g_2 f(g_2^{-1} g_1^{-1} m) = ((g_1 g_2) \star f)(m)$$

∎

64. Nota. Esto que acabamos de hacer se puede generalizar para K-homomorfismos del KG-módulo M al KG-módulo N ($\mathrm{Hom}_K(M,N)$) definiendo $\star$ exactamente de la misma manera (pues $G \hookrightarrow KG$ mandando $g \mapsto 1g$)

Ahora que hemos visto que $\mathrm{End}_K(M)$ es un G-conjunto, podemos preguntarnos por sus puntos fijos. Resulta que $(\mathrm{End}_K(M))^G = \mathrm{End}_{KG}(M)$, vea la prueba en la página 43.

65. Definición. Definimos la **traza** como la función

$$\begin{array}{rccc} \mathrm{Tr}: & \mathrm{End}_K(M) & \to & \mathrm{End}_{KG}(M) \\ & f & \mapsto & \sum_{g\in G} g\star f \end{array}$$

Capítulo 2

Vértices y Correspondientes de Green

2.1. Módulos restringidos e inducidos

Módulos restringidos

1. Definiciones. Sean M un KG-módulo, H un subgrupo de G, y $\varphi : KH \hookrightarrow KG$ el homomorfismo de anillos dado por la inclusión.

Entonces M se puede ver como un KH-módulo izquierdo definiendo, para $\sum_i k_i h_i \in KH$ y $m \in M$,

$$\left(\sum\nolimits_i k_i h_i\right) m = \varphi\left(\sum\nolimits_i k_i h_i\right) m$$

esto se conoce como **restricción**.

Al KH-módulo obtenido de M por medio de una restricción de escalares se le denota $M{\downarrow}_H^G$ y se le conoce como el KH**-módulo restringido**.

Producto Tensorial

2. Definiciones. Sean M un R-módulo derecho, N un R-módulo izquierdo y $f : M \times N \to P$, donde P es un grupo abeliano. Decimos que la aplicación f es **biaditiva** si es aditiva en ambas variables, esto es, si

$$f(m_1 + m_2, n) = f(m_1, n) + f(m_2, n)$$
$$f(m, n_1 + n_2) = f(m, n_1) + f(m, n_2)$$

para todas las $m, m_1, m_2 \in M$, $n, n_1, n_2 \in N$.

Decimos que la aplicación f es **balanceada** si

$$f(mr, n) = f(m, rn)$$

para todas las $m \in M$, $n \in N$ y $r \in R$.

3. Definiciones. Sean M un R-módulo derecho y N un R-módulo izquierdo. Denotemos por F al R-módulo libre con base $M \times N$ y por G al submódulo de F generado por

todos los elementos de las formas

$$(m_1 + m_2, n) - (m_1, n) - (m_2, n)$$
$$(m, n_1 + n_2) - (m, n_1) - (m, n_2)$$
$$(mr, n) - (m, rn)$$

donde $m, m_1, m_2 \in M$, $n, n_1, n_2 \in N$ y $r \in R$. Definimos el **producto tensorial** de M y N sobre R como

$$T = M \otimes_R N = F/G$$

y denotamos al elemento $(m, n) + G$ de T por $m \otimes n$. Así que el elemento general de T es una suma finita de la forma

$$t = \sum_i m_i \otimes n_i.$$

donde $m_i \in M$ y $n_i \in N$. Es importante notar que la representación anterior no es necesariamente única.

4. Teorema. (Propiedad universal del producto tensorial) Sean M un R-módulo derecho, N un R-módulo izquierdo, y F un R-módulo libre con base $M \times N$. Cada aplicación biaditiva y R-balanceada puede ser factorizada a través del producto tensorial $M \otimes N$.

Vea la demostración en [Ash, 2000] secc 8.7.2 p.22.

5. Proposición. Sean M, N, P R-módulos. El producto tensorial tiene las siguientes propiedades:

1. $M \otimes N \cong N \otimes M$
2. $M \otimes (N \otimes P) \cong (M \otimes N) \otimes P$
3. $M \otimes (N \oplus P) \cong (M \otimes N) \oplus (M \otimes P)$
4. Si R es un anillo, visto como R-módulo derecho, $R \otimes_R M \cong M$
5. $R^n \otimes M \cong M^n$
6. $R^m \otimes R^n \cong R^{mn}$

Las demostraciones se pueden encontrar en [Ash, 2000] secc 8.7 p. 23.

Módulos Inducidos

6. Definición. Sean M un KH-módulo y G un grupo que contenga a H. El KG-módulo **inducido** por M, denotado $M{\uparrow}_H^G$, se define por

$$M{\uparrow}_H^G = KG \otimes_{KH} M$$

7. Observación. De acuerdo con la definición de producto tensorial, los elementos de $M{\uparrow}_H^G$ son sumas finitas de la forma $\sum r_i \otimes m_i$, donde r_i está en el KG-módulo *derecho* KG y m_i en el KH-módulo *izquierdo* M. Si $r \in KG$ es de la forma $\sum k_i g_i$, las propiedades de bilinealidad y KH-balanceo del producto tensorial nos permiten escribir $r \otimes m$ como $\sum_i k_i g_i \otimes m = \sum_i 1_k g_i \otimes k_i 1_H m = \sum_i g_i \otimes m'$. Más aún, si $t = |G : H|$ y $\mathfrak{G} = \{\mathfrak{g}_1, \ldots, \mathfrak{g}_t\}$ es un conjunto completo de representantes de las

clases laterales izquierdas de G/H entonces, para todas las $g \in G$ y $m \in M$, g se puede escribir como $\mathfrak{g}_g h_g$ y, por tanto, cualquier producto $g \otimes m$ se puede escribir $\mathfrak{g}_g \otimes h_g m = \mathfrak{g}_g \otimes m'$, donde $\mathfrak{g}_g \in \mathfrak{G}$ y $m' \in M$.

Así que es común decir que, si $\mathfrak{G} = \{\mathfrak{g}_1, \ldots, \mathfrak{g}_t\}$ es un conjunto completo de representantes de las clases laterales izquierdas de G/H entonces, los elementos de $M\uparrow_H^G$ son sumas finitas de la forma

$$\sum_i \mathfrak{g}_i \otimes m_i$$

con $\mathfrak{g}_i \in \mathfrak{G}$ y $m_i \in M$.

Hasta ahora, $M\uparrow_H^G$ es solamente un grupo abeliano libre, pero nosotros queremos verlo como KG-módulo.

Para verlo como K-espacio vectorial, necesitamos definir la multiplicación por escalares y ver que es asociativa, distributiva y que la identidad del campo fija a todos los vectores. Y para que sea KG-módulo tendrá que ser un G-conjunto cuya acción pueda convivir con la multiplicación por escalares.

Si definimos

$$k(\sum_i \mathfrak{g}_i \otimes m_i) = \sum_i \mathfrak{g}_i \otimes (km_i)$$

y

$$g(\sum_i \mathfrak{g}_i \otimes m_i) = \sum_i (g\mathfrak{g}_i) \otimes m_i = \sum_i \mathfrak{g}_{\sigma(i)} \otimes (h_i m_i) = \sum_i \mathfrak{g}_{\sigma(i)} \otimes m_i'$$

tenemos las propiedades de espacio vectorial, gracias a que M es un KH-módulo izquierdo; y tenemos un G-conjunto gracias a que $\otimes$ es KH-balanceada y a que $g\mathfrak{g}_i = \mathfrak{g}_{\sigma(i)} h_i$, donde σ es la permutación que induce g al actuar por la izquierda en la clases laterales izquierdas de G.

8. Proposición. Algunas propiedades de la inducción de módulos son las siguientes:

El módulo inducido $M\uparrow_H^G$ es relativamente H libre con respecto a M. Además $M\uparrow_H^G$ es la suma directa de espacios vectoriales $M\uparrow_H^G = \sum_{s \in G/H} s \otimes M$ y cada subespacio $s \otimes v$ tiene dimensión igual a la de M.

Sean M, M_1, M_2 KH-módulos para el subgrupo H de G, y sea N un KG-módulo.

1. Si M es libre entonces $M\uparrow_H^G$ es libre;
2. Si M es proyectivo entonces $M\uparrow_H^G$ es proyectivo;
3. $(M_1 \oplus M_2)\uparrow_H^G \cong M_1\uparrow_H^G \oplus M_2\uparrow_H^G$;
4. $(M^*)\uparrow_H^G \cong (M\uparrow_H^G)^*$;
5. Si L es un subgrupo de G tal que $H \leq L \leq G$, entonces $(M\uparrow_H^L)\uparrow_L^G \cong M\uparrow_H^G$
6. $N \otimes M\uparrow_H^G \cong (N\downarrow_H^G \otimes M)\uparrow_H^G$
7. Si N^g es un KG-módulo conjugado[1] a N, entonces

$$N\uparrow_H^G \cong N^g\uparrow_{gHg^{-1}}^G$$

8. $KH\uparrow_H^G = KG$.

[1] vea la definición 19 en la página 19

Demostración: Vea las demostraciones en [Alperin, 1986] pp.56-57

$(8 \Rightarrow 5)$

$$\begin{aligned}(M\uparrow_H^L)\uparrow_L^G &= KG \otimes_{KL} (KL \otimes_{KH} M) \\ &= (KG \otimes_{KL} KL) \otimes_{KH} M \\ &= (KL\uparrow_L^G) \otimes_{KH} M \\ &= KG \otimes_{KH} M = M\uparrow_H^G\end{aligned}$$

∎

Proyectividad Relativa

9. Definición. Sean M un KG-módulo y H un subgrupo de G. Decimos que M es **libre relativo** a H si existe un KH-módulo N tal que

$$M \cong N\uparrow_H^G$$

Otra definición equivalente es la siguiente: M es relativamente H libre con respecto al KH-submódulo X de M tal que cualquier KH-homomorfismo desde X hacia cualquier KG-módulo N se extiende de manera única a un KG-homomorfismo de M a N.

10. Lema. Si M y N son KG-módulos relativamente H libres con respecto a los KH-submódulos X y Y, respectivamente, y $X \cong Y$, entonces $M \cong N$

Vea las demostraciones de este lema, la siguiente proposición y su corolario en [Alperin, 1986] pp. 54-56.

11. Proposición. Si X es un KH-módulo, donde H es un subgrupo de G, entonces existe un KG-módulo el cual es relativamente H proyectivo con respecto a X.

12. Corolario. Si M es un KG-módulo generado por el KH-submódulo X para algún subgrupo H de G, entonces M es relativamente H libre con respecto a X si y sólo si $\dim M = |G : H| \dim X$.

13. Definición. Sean M un KG-módulo y H un subgrupo de G. Decimos que M es **proyectivo relativo** con respecto a H si existe un KH-módulo N libre relativo a H tal que

$$M \mid N\uparrow_H^G$$

14. Teorema. Sean M un KG-módulo y H un subgrupo de G. Las siguientes afirmaciones son equivalentes:

1. M es un sumando directo de un módulo relativamente H libre;
2. Si φ es un homomorfismo del KG-módulo N a M y *phi* es separable como KH-homomorfismo, entonces φ es separable;
3. Si φ es un homomorfismo del KG-módulo N al KG-módulo U y ψ es un homomorfismo de M a U entonces hay un KG-homomorfismo ρ de M a N tal que $\varphi\rho = \psi$ dado que exista un KH-homomorfismo con esta propiedad.

La prueba de este teorema es análoga a la prueba del teorema equivalente para módulos proyectivos, sólo se necesita usar módulos relativamente libres en vez de módulos libres y usar la existencia del KH-homomorfismo donde se habían usado transformaciones lineales.

Vea [Alperin, 1986] pp. 56 y 29.

15. Teorema. Un KG-módulo M es proyectivo relativo con respecto a H si y sólo si $M \mid M{\downarrow}_H^G{\uparrow}_H^G$ (como KG-módulos).

Demostración: Por demostrar que esta afirmación es equivalente a cualquiera de las afirmaciones del teorema anterior (teorema 14).

Claramente, el enunciado de este teorema es una generalización del inciso (2) del teorema anterior.

Por otro lado, existe un KG-homomorfismo de $(M{\downarrow}_H^G){\uparrow}_H^G$ sobre M que es separable como KH-homomorfismo: La inclusión de $M{\downarrow}_H^G$ en M se extiende (después de hacer la identificación usual $M = 1 \otimes M$) a un KG-homomorfismo de M en $(M{\downarrow}_H^G){\uparrow}_H^G$ y éste se separa sobre H como el KH-homomorfismo de M en $(M{\downarrow}_H^G){\uparrow}_H^G$. ∎

Traza Relativa

16. Definición. Sean M un KG-módulo, H un subgrupo de G, $n = |G : H|$ y $\mathfrak{g}_1 = 1, \ldots, \mathfrak{g}_n$ tales que $G = \cup_{i=1}^n \mathfrak{g}_i H$. Definimos la **traza relativa** de H a G como la función

$$\begin{aligned} \mathrm{Tr}_{H,G} : \mathrm{End}_{KH}(M{\downarrow}_H^G) &\to \mathrm{End}_{KG}(M) \\ f &\mapsto \textstyle\sum_{i=1}^n \mathfrak{g}_i \star f \end{aligned}$$

donde $(g \star f)(m) = gf(g^{-1}m)$ para todas las $g \in G$ y $f \in \mathrm{End}_{KH}$

17. Proposición. La traza relativa $\mathrm{Tr}_{H,G}$ está bien definida, pues no depende de la elección de los representantes de las clases laterales izquierdas.

Demostración: Sea $\mathfrak{G} = \{\mathfrak{g}_1, \ldots, \mathfrak{g}_n\}$ un conjunto de representantes de las clases laterales izquierdas y $\{\mathfrak{g}_1 h_1, \ldots, \mathfrak{g}_n h_n\}$ otro conjunto de representantes. Basta probar que $(\mathfrak{g}_i h_i) \star f = \mathfrak{g}_i \star f$.

Recordemos que, como f es un KH-endomomorfismo, $f(hm) = hf(m)$ para toda $h \in H$ y $m \in M$, así que $((\mathfrak{g}_i h_i) \star f)(m) = \mathfrak{g}_i h_i f(h_i^{-1}\mathfrak{g}_i^{-1} m) = \mathfrak{g}_i h_i h_i^{-1} f(\mathfrak{g}_i^{-1} m) = \mathfrak{g}_i f(\mathfrak{g}_i^{-1} m) = (\mathfrak{g}_i \star f)(m)$ ∎

Para nosotros la utilidad de la traza relativa reside en el siguiente teorema. Véase la demostración en [Benson, 1991] pp. 70-71.

18. Teorema. (Criterio de Higman) Un KG-módulo M es proyectivo relativo con respecto a H si y sólo si el mapeo $\mathrm{id}_M \in \mathrm{Im}(\mathrm{Tr}_{H,G})$ (esto es, si la traza relativa $\mathrm{Tr}_{H,G}$ es una función suprayectiva).

2.2. Módulos Conjugados

19. Definición. Sean $H \leq G$ grupos, y N un KH-módulo. Sabemos entonces que existe una representación de H $\rho : H \to \mathrm{GL}_n(K)$. Así que para todo $g \in G$ existe una representación $\varphi : gHg^{-1} \to \mathrm{GL}_n(K)$, que resulta de componer la función $\psi : gHg^{-1} \to H$ que manda ghg^{-1} a h con ρ: $\varphi = \rho\psi$ Al KH-módulo que corresponde a esta representación φ se le denota por N^g y se conoce como el **módulo conjugado** por g obtenido a partir de N.

20. Teorema. Si M y N son KG-módulos, entonces

$$(M \oplus N)^g \cong M^g \oplus N^g$$

y

$$(M^g)^{g^{-1}} \cong M$$

2.3. Vértices y Fuentes

Para toda esta sección ocupamos un KG-módulo inescindible M, y p denota la característica del campo K.

Vértices

21. Definición. Sea M un KG-módulo inescindible. Un vértice de M es un subgrupo de G con la siguiente propiedad:

1. M es proyectivo relativo con respecto a D;
2. M no es proyectivo relativo con respecto a ningún subgrupo propio de D.

22. Teorema. Sean D_1 y D_2 dos vértices de M, y sean N_1 y N_2 las fuentes que corresponden, respectivamente, a dichos vértices. Entonces D_1 y D_2 son G-conjugados; y, también, N_1 y N_2 son G-conjugados

23. Corolario. Si N es una fuente de M que corresponde al vértice D, entonces

$$N{\uparrow}_H^G \cong N^g{\uparrow}_{gHg^{-1}}^G$$

y, por lo tanto, M es sumando directo de la descomposición en suma de inescindibles de ambos.

24. Teorema. Si D es un vértice de M, entonces D es un p-subgrupo de G.

25. Teorema. M es proyectivo relativo para todo subgrupo H de G que contenga al p-subgrupo de Sylow de G

Demostración: Basta probar que

$$\mathrm{Tr}_{G,H} : \mathrm{End}_{KH}(M{\downarrow}_H^G) \to \mathrm{End}_{KG}(M)$$

es suprayectiva. Claramente $1_M \in \mathrm{End}_{KH}(M{\downarrow}_H^G)$ y, como p y $[G:H]$ son primos relativos, existe $k \in K$ tal que $k[G:H] = 1$. Como End_{KH} es un K-espacio vectorial, $k1_M \in \mathrm{End}_{KH}$ y $\mathrm{Tr}_{G,H}(k1_M) = 1_M$ pues, si $\{g_1H, \ldots, g_{[G:H]}H\}$ son las clases laterales izquierdas de H en G, y $m \in M$, entonces

$$\mathrm{Tr}_{G,H}(k1_M)(m) = \sum_{i=1}^{[G:H]} g_i(k1_M)(g_i^{-1}m) = \sum_{i=1}^{[G:H]} g_i k g_i^{-1} m = k[G:H]m = m$$

Por lo tanto, la preimagen de 1_M bajo $\mathrm{Tr}_{G,h}$ es no vacía, y la función es suprayectiva.

Otra demostración se encuentra en [Alperin, 1986] p.65. ∎

26. Teorema. Sea M un KG-módulo inescindible y proyectivo relativo con respecto a H.

1. entonces es proyectivo relativo con respecto a gHg^{-1} para todo $g \in G$;

2. y si, $H \leq X \leq G$ es otro subgrupo, entonces M es proyectivo relativo con respecto a X.

En conclusión, no tiene mucho mérito ser proyectivo relativo con respecto a un subgrupo grande con respecto a G.

Demostración: Una manera de demostrar este teorema es ajustar la demostración del teorema anterior y ver que $\mathrm{Tr}_{G,gHg^{-1}}$ y $\mathrm{Tr}_{G,X}$ son suprayectivas. Otra manera es la siguiente:

2. Por hipótesis existe un KH-módulo N tal que $M \mid N{\uparrow}_H^G$. Pero $N{\uparrow}_H^G = (N{\uparrow}_H^X){\uparrow}_X^G$. Claramente $U = N{\uparrow}_H^X$ es un KX-módulo tal que $M \mid U{\uparrow}_X^G$.

∎

Fuentes

Para toda esta sección ocupamos un KG-módulo inescindible M, p denota la característica del campo K y D denota a un vértice de M.

27. Definición. El KD-módulo restringido $M{\downarrow}_D^G$ se puede descomponer como suma la directa $\oplus_i M_i$ de KD-módulos inescindibles, y resulta que existe i_0 tal que el KG-módulo M es sumando directo del KG-módulo inducido $M{\uparrow}_D^G$. A este KD-módulo M_{i_0} se le conoce como la **fuente** correspondiente al vértice D.

28. Nota. En este trabajo no utilizamos para nada las fuentes, la definición está aquí solamente porque es costumbre definirlas cada vez que se definen los vértices.

2.4. Correspondencia de Green

Sean Q un p-subgrupo de G y L un subgrupo que contenga a $N_G(Q)$. Si P y R son subgrupos de G, escribimos $P \leq_G R$ para indicar que un conjugado de P está contenido en R. Si $\mathscr{H}$ es una colección de subgrupos de G, entonces $P \leq_G \mathscr{H}$ significa que $P \leq_G H$ para alguna $H \in \mathscr{H}$. Ahora fijaremos algunas colecciones de p-subgrupos de G. Sean

$$\begin{aligned}
\mathfrak{x} &= \{sQs^{-1} \cap Q : s \in G, s \notin L\} \\
\mathfrak{y} &= \{sQs^{-1} \cap L : s \in G, s \notin L\} \\
\mathfrak{z} &= \{R : R \leq Q, R \not\leq_G \mathfrak{x}\}
\end{aligned}$$

Uno debería pensar que $\mathfrak{x}$ y $\mathfrak{y}$ consisten de los subgrupos "pequeños", con respecto a Q (aunque $\mathfrak{y}$ bien pudiera contener un conjugado de Q) y que $\mathfrak{z}$ consiste de Q y sus subgrupos "grandes".

Note que cada subgrupo de $\mathfrak{x}$ es un subgrupo propio de Q, como $L \supseteq N_G(Q)$, también $Q \in \mathfrak{z}$. Finalmente, si $\mathscr{H}$ es cualquier colección de subgrupos de G diremos que el KG-módulo M es proyectivo relativo con respecto a $\mathscr{H}$ si M es suma directa de módulos cada uno de los cuales es proyectivo relativo para un subgrupo de $\mathscr{H}$. Ahora podemos enunciar el resultado fundamental, vea la demostración en [Alperin, 1986] pp.80-81.

29. Teorema. (Correspondencia de Green) Existe una correspondencia uno a uno entre las clases de isomorfismo de KG-módulos inescindibles con vértice en $\mathfrak{z}$ y las clases de isomorfismo de KL-módulos inescindibles con vértice en $\mathfrak{z}$. Si M y N son tales módulos para G y L, respectivamente, los cuales se corresponden entonces M y N tienen el mismo vértice y

$$M{\downarrow}_L^G \cong N \oplus Y$$
$$N{\uparrow}_L^G \cong M \oplus X$$

donde Y es un KL-módulo proyectivo relativo con respecto a $\mathfrak{y}$ y X es un KG-módulo proyectivo relativo con respecto a $\mathfrak{x}$.

Para probar la correspondencia de Green requerimos de los siguientes resultados:

30. Lema. Si M es un KG-módulo inescindible con vértice Q y H es un subgrupo de G que contiene a Q, entonces hay un KH-módulo inescindible N que satisface cualesquiera dos de las siguientes afirmaciones:

1. N es sumando directo de $M{\downarrow}_H^G$
2. M es sumando directo de $N{\uparrow}_H^G$
3. N tiene vértice Q.

Demostración en [Alperin, 1986] p.68.

31. Lema. Si N es un KH-módulo proyectivo relativo con respecto a Q, donde Q es un subgrupo del subgrupo H de G, entonces $(N{\uparrow}_H^G){\downarrow}_H^G \cong N \oplus W$ donde todo sumando inescindible de W es proyectivo relativo para un subgrupo de la forma $sQs^{-1} \cap H$, $s \in G$ y $s \notin H$.

Demostración en [Alperin, 1986] p. 69.

32. Lema. Si R es un subgrupo de Q, entonces las siguientes afirmaciones son equivalentes:

1. $R \subseteq_G \mathfrak{x}$
2. $R \subseteq_L \mathfrak{x}$
3. $R \subseteq_L \mathfrak{y}$

Demostración: (1 implica 2) Si (1) es cierto, entonces existe $g \in G$ tal que $gRg^{-1} \subseteq sQs^{-1} \cap Q$, donde $s \in G$ y $s \notin L$. Si $g \in L$, claramente tenemos (2); y si $g \notin L$, entonces $R \subseteq g^{-1}Qg$ y, como $R \subseteq Q$, $R \subseteq g^{-1}Qg \cap Q$, esto es, R está en $\mathfrak{x}$, así que también tenemos (2).

(2 implica 3) Si (2) es cierto, entonces tenemos $x \in L$ tal que $xRx^{-1} \subseteq sQs^{-1} \cap Q$ para alguna $s \in G$ tal que $s \notin L$. Como $Q \subseteq N_G(Q) \subseteq L$, tenemos que $xRx^{-1} \subseteq sQs^{-1} \cap L$ y, por lo tanto, $R \subseteq_G \mathfrak{y}$.

(3 implica 1) Finalmente, supongamos que (3) es cierto. Entonces existen $x \in L$, y $s \in G$ tal que $s \notin L$ tales que $xRx^{-1} \subseteq sQs^{-1} \cap L$. Así que $R \subseteq (x^{-1}s)Q(s^{-1}x) \cap L$, pues $x \in L$. Pero $x^{-1}s \notin L$ (pues si lo estuviera, entonces s estaría en L), así que tenemos $R \subseteq_G \mathfrak{x}$ en la forma de $R \subseteq Q \cap (x^{-1}s)Q(s^{-1}x)$. ∎

33. Definición. Si H es una colección de subgrupos de G, decimos que el KG-módulo M es proyectivo relativo con respecto a H si M es suma directa de módulos cada uno de los cuales es proyectivo relativo con respecto a un subgrupo de H.

34. Lema. Si M es un KG-módulo proyectivo relativo con respecto a $\mathfrak{x}$, entonces $M{\downarrow}_L^G$ es proyectivo relativo con respecto a $\mathfrak{y}$. Si N es un KL-módulo proyectivo relativo con respecto a Q y con respecto a $\mathfrak{y}$, entonces $N{\uparrow}_H^G$ es proyectivo relativo con respecto a $\mathfrak{x}$.

Demostración: Sea W un sumando inescindible de M. Así que W es proyectivo relativo para un subgrupo de la forma $sQs^{-1} \cap Q$, con $s \in G$ tal que $s \notin L$. Entonces, por el teorema de Mackey, $W{\downarrow}_L^G$ es proyectivo relativo para la colección de subgrupos de la forma $t(sQs^{-1} \cap Q)t^{-1} \cap L = tsQs^{-1}t^{-1} \cap tQt^{-1} \cap L$. Pero o $t \notin L$ o $ts \notin L$ así que dicho subgrupo está contenido en un elemento de $\mathfrak{y}$ y por lo tanto $W{\downarrow}_L^G$ y $M{\downarrow}_L^G$ son proyectivos relativos con respecto a $\mathfrak{y}$.

Por otro lado, si W sumando inescindible de N y W tiene vértice P entonces $P \subseteq_L \mathfrak{y}$ así que $W{\uparrow}_H^G$ es proyectivo relativo con respecto a P y $P \subseteq_G \mathfrak{x}$, por el Lema 32, $W{\uparrow}_H^G$ es proyectivo relativo con respecto a $\mathfrak{x}$ y el lema está probado. ∎

35. Lema. Si M es un KG-módulo inescindible con vértice $R \in \mathfrak{z}$, entonces $M{\downarrow}_L^G \cong N \oplus Y$, donde N es un KL-módulo inescindible con vértice R, M es sumando directo de $N{\uparrow}_H^G$ y Y es un KL-módulo proyectivo relativo con respecto a $\mathfrak{y}$.

Demostración: Por el lema 30, hay un KL-módulo inescindible N con vértice R y M es un sumando directo de $N{\uparrow}_H^G$. Ahora $(N{\uparrow}_H^G)_L \cong N \oplus Y_1$, donde Y_1 es proyectivo relativo con respecto a $\mathfrak{y}$, por el lema 31, $M{\downarrow}_L^G$ es ya sea isomorfo a $N \oplus Y$ o es isomorfo a Y para algun sumando Y de Y_1. Sin embargo, de nuevo por el lema 30, $M{\downarrow}_L^G$ tiene un sumando inescindible W con vértice R. Ahora W no puede ser isomorfo con un sumando de Y, o si no $R \subseteq_L \mathfrak{y}$ y $R \subseteq_G \mathfrak{x}$, por el lema 32, $R \notin \mathfrak{z}$. Por lo tanto, $W \cong N$ y $M{\downarrow}_L^G \cong N \oplus Y$, como se afirmaba. ∎

36. Lema. Si N es un KL-módulo inescindible con vértice $R \in \mathfrak{z}$ entonces $N{\uparrow}_H^G \cong M \oplus X$ donde M en un KG-módulo inescindible con vértice R, N es un sumando directo de $M{\downarrow}_L^G$ y X es un KG-módulo proyectivo relativo con respecto a $\mathfrak{x}$.

Demostración: Sea $N{\uparrow}_H^G = M_1 \oplus \cdots \oplus M_r$ una suma directa de KG-módulos inescindibles. Como $(N{\uparrow}_H^G)_L \cong N \oplus Y$, donde Y es un KL-módulo proyectivo relativo con respecto a $\mathfrak{y}$, por el lema 31, tenemos, después de una renumeración, que $(M_1)_L \cong N + Y_1$, $(M_i)_L \cong Y_i$, para $i = 2, \ldots, r$, donde los Y_i son KL-módulos y $Y \cong Y_1 \oplus \cdots \oplus Y_r$. Afirmamos que M_1 tiene un vértice en $\mathfrak{z}$ y que $M_2, \ldots, M_r$ son relativos proyectivos con respecto a $\mathfrak{x}$. De hecho, M_1 tiene un vértice en $\mathfrak{z}$, ya que M_1 es sumando directo de $N{\uparrow}_H^G$, y M_1 no puede ser proyectivo relativo con respecto a $\mathfrak{x}$, ya que el lema 34 implicaría que $(N_1)_L \cong N \oplus Y_1$ es proyectivo relativo con respecto a $\mathfrak{y}$, que no es el caso. Por lo tanto M_1 tiene vértice en $\mathfrak{z}$. Además si M_i, $i > 1$ no fuera proyectivo relativo con respecto a $\mathfrak{x}$, entonces el lemma 35 se le aplicaría i $(M_i)_L$ no sería proyectivo relativo con respecto a $\mathfrak{y}$, lo cual sí es. Por lo tanto M es de hecho, proyectivo relativo con respecto a $\mathfrak{x}$.

Haciendo $M = M_1$, $X = M_2 + \cdots + M_r$, tenemos que M es sumando directo de $N{\uparrow}_H^G$, $N{\uparrow}_H^G \cong M \oplus X$, donde X es proyectivo relativo con respecto a $\mathfrak{x}$. Sólo queda probar que M tiene vértice R sin embargo, el vértice de M está en $\mathfrak{z}$. Entonces se aplica el lema 35 y existe un sumando único en cualquier descomposición de $M{\downarrow}_L^G$ en la suma directa de módulos inescindibles lo cual no es proyectivo relativo con respecto a $\mathfrak{y}$ y ese módulo tiene un vértice igual a un vértice de M, pero $M{\downarrow}_L^G \cong N \oplus Y$ por lo tanto M tiene vértice R como N tiene vértice R. Esto prueba el lema. ∎

Para demostrar la correspondencia de Green (teorema 29), vemos que los lemas 35 y 36 prueban todo excepto que la correspondencia es uno a uno. Para ello necesitamos

probar dos cosas: Si M es un KG-módulo inescindible con vértice R en $\mathfrak{z}$, N es como en el lema 35, y $N{\uparrow}_H^G = M' \oplus Y$ como en el lema 36, entonces $M \cong M'$; si comenzamos con N vale un resultado análogo. Sin embargo, en el lema 35 probamos que M es sumando directo de $N{\uparrow}_H^G$ y en el lema 36 que N es sumando directo de M_L, así que hemos terminado.

Capítulo 3

Pesos y Conjetura de Alperin

En este capítulo solamente intentamos presentar las definiciones y enunciar los resultados necesarios para nuestros propósitos. En los libros [Alperin, 1986] y [Curtis and Reiner, 1990] y en el artículo [Alperin, 1987] el lector interesado podrá encontrar información ampliada sobre el tema.

3.1. Pesos

Para esta sección asumiremos que G es un grupo finito; que K es un campo finito de característica p; y que p divide al orden de G.

En su artículo [Alperin, 1987], Alperin define un peso de la siguiente manera.

1. Definición. 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.

2. Definición. Dos pesos (Q, S) y (Q', S') se dicen **equivalentes** si existe un elemento $g \in G$ tal que

   1. $Q' = gQg^{-1}$;
   2. $S' = S^g$.

3. Teorema. (Burry-Carlson-Puig) Sean $Q \leq H \leq G$ tales que Q es un p-grupo y $N_G(Q) \leq H$. Suponga que M es un KG-módulo inescindible tal que $M{\downarrow}_H^G$tiene un componente N con vértice Q. Entonces Q es un vértice de M y N es el correspondiente de Green de M con respecto a (G, Q, H)

4. Proposición. Sean G un grupo finito, K un campo algebraicamente cerrado de característica p, (Q, S) un peso para KG, H un subgrupo de G tal que $Q \leq H$ y S tiene una cantidad diferente de cero de puntos fijos en $H \cap N_G(Q)$ (por ejemplo, H puede ser un p-subgrupo de G que contenga a Q). Entonces el Correspondiente de Green de S es un sumando directo de $K{\uparrow}_H^G$.

Demostración: Como S tiene vértice Q, es una consecuencia del teorema de Burry-Carlson-Puig que para mostrar que el correspondiente de Green de S es un sumando de $K(G/H)$ basta probar que S es un sumando de $K(G/H)\downarrow^G_{N_G(Q)}$.

Hagamos $L = N_G(Q)$ y $R = H \cap L$. Por el teorema de Mackey

$$K(G/H)\downarrow^G_L = K\uparrow^G_H\downarrow^G_L \cong \otimes_{LgH}({}^gK)\downarrow^{{}^gH}_{L\cap {}^gH}\uparrow^L_{L\cap {}^gH}$$

En particular, cuando $g = 1$, el módulo de permutación $K\downarrow^H_R\uparrow^L_R = K\uparrow^L_R$ es un sumando de $K(G/H)\downarrow^G_L$. Sin embargo, eso es simplemente el módulo de permutación $M = K\uparrow^{L/Q}_{R/Q}$ visto como un KL-módulo donde Q actúa trivialmente. Basta probar que S es un sumando directo de M como KL-módulos o, de manera equivalente, como $K(L/Q)$-módulos (ya que Q actúa trivialmente tanto en S como en M).

Como S es un módulo proyectivo para $K(L/Q)$, por hipótesis, es suficiente mostrar que S es una imagen isomorfa de M para mostrar que S es un sumando de M. Pero S es también un KL-módulo simple, así que sólo necesitamos probar que $\mathrm{Hom}_{K(L/Q)}(M, S) \neq 0$. Sin embargo, por el teorema de reciprocidad de Frobenius tenemos:

$$\begin{aligned}\mathrm{Hom}_{K(L/Q)}(M, S) &= \mathrm{Hom}_{K(L/Q)}(K\uparrow^{L/Q}_{R/Q}, S)\\ &\cong \mathrm{Hom}_{K(R/Q)}(K, S\downarrow^{L/Q}_{R/Q}) \neq 0\end{aligned}$$

∎

3.2. La conjetura de Alperin

5. 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.

6. Nota. 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.

7. Nota. En el artículo [Valero-Elizondo, 2002] hay un resumen actualizado de lo que sabemos hasta ahora de la conjetura de Alperin y sus consecuencias:

Esta versión de la conjetura implica la original, que puede ser obtenida sumando las igualdades de la conjetura fuerte sobre los bloques. Esta conjetura fuerte ha sido probada para cuando G es un:

- Grupo finito de Lie y característica p [Cabanes, 1988].
- Grupo soluble [Okuyama].

- Grupo simétrico [Alperin and Fong, 1990].
- $GL(n, q)$ y p no divide a q [Alperin and Fong, 1990].

La prueba de Alperin y Fong en el caso de los grupos simétricos fue simplemente una observación de una igualdad numérica que no sugiere una razón más profunda por la cual se de la relación. Para grupos finitos en general uno no espera tener ninguna relación canónica entre los pesos y los módulos simples; de hecho, Alperin mismo dice que esto no es muy probable (Vea [Alperin, 1987] p. 369). Para grupos de Lie existe una biyección canónica. Como los grupos simétricos y los grupos de Lie tienen fuertes conexiones en la teoría de sus representaciones, es razonable preguntarse si existe una biyección canónica para el caso de los grupos simétricos.

Si la conjetura de Alperin fuera cierta, implicaría un número de resultados conocidos, que hasta ahora se consideran no relacionados (vea [Alperin, 1987]). También es razonable esperar que si se puede dar una biyección explícita para probarla, esta pudiera revelar nuevas conexiones entre los KG-módulos simples y los pesos; hay muchos resultados acerca de los primeros, y los segundos están relacionados con los bloques de defecto cero, los cuales no son tan fáciles de manejar como los módulos simples. De hecho, la verdadera importancia de la conjetura de Alperin es que provee una conexión entre los bloques de defecto cero y el conjunto de todos los módulos simples. Mas específicamente, [Knörr and Robinson, 1989] ha probado que la conjetura de Alperin es equivalente a una afirmación la cual expresa el número de bloques de defecto cero de un grupo en términos del número de irreducibles p-modulares de secciones del grupo de la forma $N_G(P)/P$, donde $P \leq G$ es un p-subgrupo. Estos últimos números son fáciles de calcular, ya que por un teorema de Brauer, el número de irreducibles p-modulares de un grupo es igual al número de clases de conjugación p-regulares.

Capítulo 4

Experimentos computacionales acerca de la conjetura de Alperin

4.1. Objetivos y estructura general del programa

La conjetura de Alperin normalmente se enuncia pidiendo que K sea un campo algebraicamente cerrado. Hasta donde el autor y su asesor saben, se desconoce si esta hipótesis es en realidad necesaria o si la conjetura es válida aún con una hipótesis más débil sobre K.

El objetivo de este trabajo es explorar la conjetura de Alperin en el caso en que K es un campo finito (y, por lo tanto, *NO* es algebraicamente cerrado) y G es un grupo finito. Para ello creamos rutinas en GAP [The GAP Group, 2002] que nos permitan comprobar la conjetura para algunos ejemplos. Esto es, rutinas que nos permitan contar el número de KG-módulos simples (no isomorfos entre sí) y el número de pesos (no conjugados entre sí) de KG.

El método que utilizaremos será sumamente simple: Por un lado contamos el número de KG-módulos simples; por otro lado, calculamos el número de pesos para KG; y, finalmente, comparamos ambos números.

Comencemos presentando la estructura general del programa: [1]

⟨ *⟩≡
⟨ *Generating KG-modules*⟩
⟨ *Decomposing KG-modules*⟩
⟨ *Meat Axe and simple KG-modules*⟩
⟨ *Green correspondence*⟩
⟨ *Alperin conjecture*⟩

[1] Todo el código esta escrito en inglés para facilitar su comprensión a la comunidad internacional

Como sugiere esta parte del código, las rutinas se pueden clasificar en cuatro categorías:

1. Rutinas para crear estructuras de datos que representen KG-módulos;
2. rutinas para descomponer KG-módulos como suma de inescindibles;
3. rutinas para verificar si un KG-módulo es simple, que dependen del algoritmo Meat Axe ;
4. rutinas para encontrar correspondientes de Green; y
5. rutinas para verificar la conjetura de Alperin.

4.2. Generando KG-módulos

⟨ *Generating KG-modules* ⟩≡

```
    # Object Oriented Functions
    ⟨ DeclareKGObject ⟩

    # Generating KG-modules
    ⟨ KGEmptyModule ⟩
    ⟨ KGActionMatrix ⟩
    ⟨ KGTrivialModule ⟩
    ⟨ KGRegularModule ⟩
    ⟨ KGPermutationModule ⟩
    ⟨ KGRestrictedModule ⟩
    ⟨ KGInducedModule ⟩
```

KG-módulos en GAP

En este momento estamos interesados en definir una estructura de datos adecuada para representar los KG-módulos. Como el algoritmo de Meat Axe ya está implementado en GAP , y queremos usar algunas de sus rutinas, nos interesa que nuestras estructuras sean compatibles con las que entiende Meat Axe .

Encontramos que para el Meat Axe implementado en GAP un KG-módulo es una estructura que guarda el campo K (`field`), la dimensión del módulo (`dimension`), y las matrices que representan la acción del grupo (`generators`), junto con algunos objetos de uso interno, entre ellos la bandera `isMTXModule` y la lista `smashMeatAxe`.

Para mantener la compatibilidad, nuestra estructura debe incluir todo esto más los datos adicionales que nos interesen. Hemos agregado al grupo G, su identificador en GAP y una lista de generadores del grupo. Esto se guarda en `group`, `group_id` y `group_generators`, respectivamente.

Por último, note que normalmente tenemos dos versiones de la misma función, una termina con `NC` y la otra no. `NC` significa "no check" y esta versión no realiza ninguna comprobación en sus argumentos, mientras que la versión sin `NC` revisa todos sus argumentos antes de intentar trabajar con ellos.

⟨*KGEmptyModule*⟩≡

```
# KGEmptyModule
#
# The skeleton of a MeatAxe-compatible KG-module,
#
# After calling this function and have the Module's dimension
# and generators set, make them immutable.
#
# Version 0.2
# Last modified September 12th, 2004.
# Acme

KGEmptyModuleNC := function(K,G)
local M;

  M := rec(
    field := K,
    group := G,
    group_id := IdGroup(G),
    generators := [],
    dimension := [],
    isMTXModule := true,
    isKGModule := true
  );

  if Size(G) = 1 then
    # We want all groups to have generators,
    # so, for the trivial group, we want to
    # use its only element as generator.
    M.group_generators := [One(G)];
    M.smashMeataxe := rec( IsZeroGens:=true );
  else
```

```
    M.group_generators := SmallGeneratingSet(G);
  fi;

  MakeImmutable(M.field);
  MakeImmutable(M.group);
  MakeImmutable(M.group_id);
  MakeImmutable(M.group_generators);
  MakeImmutable(M.isMTXModule);
  MakeImmutable(M.isKGModule);

  return M;
end;

KGEmptyModule := function(K,G)
  if not IsField(K) then
    return Error("K must be a Field");
  elif not IsGroup(G) then
    return Error("G must be a Group");
  else
    return KGEmptyModuleNC(K,G);
  fi;
end;
```

En la página 7 vimos que un KG-módulo M se puede ver como una estructura de K espacio vectorial que es al mismo tiempo un G-conjunto y que basta conocer la acción en un conjunto de generadores de G para conocerla en todo M. La manera en que guardamos la acción de los generadores de G es usando representaciones matriciales.

1. Definición. Sea M un KG-módulo. Una representación para M es un homomorfismo $\rho : G \to \mathrm{GL}(M)$ del grupo G en el grupo de las transformaciones lineales de M visto como K espacio vectorial. Si la dimensión del espacio vectorial M es n, podemos reemplazar $\mathrm{GL}(M)$ por $\mathrm{GL}_n(K)$ y, en este caso, decimos que ρ es una representación matricial.

2. Observación. Para recuperar al KG-módulo M de su representación usamos que

$$gm = \rho(g)(m)$$

`KGActionMatrix` devuelve, para cada `s` $\in$ `S`, la representación matricial $\rho(s)$.

⟨*KGActionMatrix*⟩≡

```
# KGActionMatrix
#
# Given a KG-module M and S subset G, for each s in S,
# KGActionMatrix returns a matrix A_s such that for all m in M,
# gm = A_sm.
# In other words, if \rho:G -> GL_n(K) is a matrix
# representation of M, KGActionMatrix returns \rho(s) for each
# s in S.
#
# Version 0.2
# Last modified June 7th, 2004.
# Acme

KGActionMatrixNC := function(M, S)
  local rho, c, Reps;

  rho := GroupHomomorphismByImages(M.group,
  GL(M.dimension,Size(M.field)), M.group_generators,
  M.generators);

  Reps := [];
  for c in [1..Size(S)] do
    Reps[c] := Image(rho,S[c]);
  od;

  return Reps;
end;

KGActionMatrix := function(M, S)
  if IsSubset(M.group,S) then
    return KGActionMatrixNC(M, S);
  else
    Error("KGActionMatrix: Error. S is not a subset of G.\n");
  fi;
end;
```

El objeto KG

Siguiendo el estilo de programación usual en GAP , creamos un objeto llamado KG. Éste contiene todas las funciones que se le pueden aplicar a (una estructura de datos que represente) un KG-módulo.

En particular, los KG-módulos en GAP tienen atributos, es decir, funciones que nos dicen algunas características del KG-módulo. Estas características son calculadas una sola vez y almacenadas dentro de la estructura de datos, de tal manera que la siguiente vez que preguntemos lo mismo ya no es necesario hacer cálculos. Algunos ejemplos de atributos son: El campo K; el grupo G; si el módulo es o no simple; si es o no inescindible; etc.

⟨*DeclareKGObject*⟩≡

```
DeclareInfoClass("InfoKGModules");

KG :=rec(name:="KG-Modules");

KG.Attribute := function(arg)
  local string, real_work_getter;

  if (Length(arg)=0) or (Length(arg)>2) then
    return Error("Usage: KG.Attribute <string> [real_work_getter]");
  fi;

  if Length(arg) = 2 then
    real_work_getter := arg[2];
  else
    real_work_getter := ReturnFail;
  fi;
  string := arg[1];

  return function(module)
    if not IsBound(module.(string)) then
      module.(string) := real_work_getter(module);
    fi;
    return module.(string);
  end;
end;

KG.Field := KG.Attribute("field");
KG.Group := KG.Attribute("group");
KG.GroupID := KG.Attribute("group_id");
KG.GroupGenerators := KG.Attribute("group_generators");
KG.Dimension := KG.Attribute("dimension");
KG.Generators := KG.Attribute("generators");
KG.IsKGModule := KG.Attribute("isKGModule");
KG.Size := KG.Attribute("size", function(M)
  return Size(M.field)^M.dimension;
end);
# TODO: Uncomment
# KG.IsIdecomposable := KG.Attribute("isIdecomposable",
#                       KG.TestIsIdecomposable);
```

```
KG.IsIrreducible := MTX.IsIrreducible;
KG.IsAbsolutelyIrreducible := MTX.IsAbsolutelyIrreducible;
# TODO: Set-up an isFree attribute
# TODO: Set-up a Rank attribute

KG.Setter := function(string)
  return function(module, obj)
    if not IsBound(module.KG) then
      module.KG := rec();
    fi;
    module.KG.(string) := obj;
  end;
end;

KG.Getter := function(string)
  return function(module)
    if ( not IsBound(module.KG) ) or
       ( not IsBound(module.KG.(string)) ) then
      return fail;
    else
      return module.KG.(string);
    fi;
  end;
end;

KG.Tester := function(string)
  return function(module)
    return IsBound(module.KG) and IsBound(module.KG.(string));
  end;
end;

KG.MinimalRelativelyProjectiveSubgroup :=
KG.Getter("minimal_relatively_projective_subgroup");

KG.SetMinimalRelativelyProjectiveSubgroup :=
KG.Setter("minimal_relatively_projective_subgroup");

KG.HasMinimalRelativelyProjectiveSubgroup :=
KG.Tester("minimal_relatively_projective_subgroup");
```

Algunos KG-módulos

Aquí definimos a los KG-módulos: trivial, regular, de permutación, restringido e inducido.

⟨*KGTrivialModule*⟩≡

```
# KGTrivialModule
#
# Returns a KG module isomorphic to K
# (one which gm = m for every m)
#
# Version 0.2
# Last modified September 12th, 2004.
# Acme

KGTrivialModule := function( K, G )
  local M, c;

  M := KGEmptyModule(K,G);
  M.dimension := 1;

  M.permutations := [];
  for c in [1..Size(M.group_generators)] do
    M.generators[c] := [[One(K)]];
    M.permutations[c] := ();
  od;
  M.IsPermutationModule := true;

  MakeImmutable(M.dimension);
  MakeImmutable(M.generators);
  MakeImmutable(M.permutations);
  MakeImmutable(M.IsPermutationModule);

  return M;
end;
```

⟨*KGRegularModule*⟩≡

```
# KGRegularModule
#
# Given a field K and a group G, returns the
# group algebra KG
#
# Version 0.2
# Last modified September 12th, 2004.
# Acme

KGRegularModule := function(K,G)
  local hom, M, c, h;

  if Size(G) = 1 then
    return KGTrivialModule(K,G);
  else
    M := KGEmptyModule(K,G);
    M.dimension := Size(G);

    hom := ActionHomomorphism(G, AsList(G), OnRight);

    M.permutations := [];
    for c in [1..Size(M.group_generators)] do
      M.permutations[c] := Image(hom,M.group_generators[c]);
      M.generators[c] := PermutationMat(M.permutations[c],
      M.dimension, K);
    od;
    M.IsPermutationModule := true;

    MakeImmutable(M.dimension);
    MakeImmutable(M.generators);
    MakeImmutable(M.permutations);
    MakeImmutable(M.IsPermutationModule);

    return M;
  fi;
end;
```

⟨*KGPermutationModule*⟩≡

```
# KGPermutationModule
#
# Given a field K, a group G and a subgroup H of G, generate
# the KG-module with basis the set of right cosets of G/H.
#
# Version 0.2
# Last modified September 12th, 2004.
# Acme

KGPermutationModuleNC := function(K, G, H)
  local cosets, M, hom, c;

  cosets := RightCosets(G, H);

  if Size(cosets) = 1 then
    return KGTrivialModule(K,G);
  else
    M := KGEmptyModule(K,G);
    M.dimension := Size(cosets);

    hom := ActionHomomorphism(G, AsList(cosets), OnRight);

    M.permutations := [];
    for c in [1..Size(M.group_generators)] do
      M.permutations[c] := Image(hom, M.group_generators[c]);
      M.generators[c] := PermutationMat(M.permutations[c], M.dimension, K);
    od;
    M.IsPermutationModule := true;

    MakeImmutable(M.dimension);
    MakeImmutable(M.generators);
    MakeImmutable(M.permutations);
    MakeImmutable(M.IsPermutationModule);

    return M;
  fi;
end;

KGPermutationModule := function(K, G, H)
  if ( not IsSubgroup(G,H) ) then
    Error("KGPermutationModule: Error: H is not a subgroup of G\n");
  fi;
  return KGPermutationModuleNC(K, G, H);
end;
```

Los KG-módulos restringido e inducidos se definieron en la sección 2.1, página 15.

⟨*KGRestrictedModule*⟩≡

```
# KGRestrictedModule
#
# Regard a KG-module as a KH-module
#
# Version 0.2
# Last modified September 12th, 2004.
# Acme

KGRestrictedModuleNC := function(KG, H)
  local KH, F, hom, hom2, c, word;

  KH := KGEmptyModule(KG.field,H);
  KH.dimension := KG.dimension;

  KH.generators := KGActionMatrix( KG, KH.group_generators );

  MakeImmutable(KH.dimension);
  MakeImmutable(KH.generators);

  return KH;
end;

KGRestrictedModule := function(KG, H)
  # Verify that H is really a subgroup of G
  if ( not IsSubgroup(KG.group,H) ) then
    Error("KGRestrictedModule: Error: H is not a subgroup of G\n");
  fi;
  return KGRestrictedModuleNC(KG, H);
end;
```

⟨*KGInducedModule*⟩≡

```
# KGInducedModule
#
# Given a KH-module M and a group G containing H
# return the induced KG-module
#
# Version 0.2
# Last modified September 13th, 2004.
# Acme

KGInducedModuleNC := function(M, G)
  local N, S, blocks, list, i,j,k, g, h, H, m, mat;

  N := KGEmptyModule(M.field, G);
  N.dimension := Size(G)/Size(M.group)*M.dimension;

  H := M.group;
  S := RightTransversal(G,H);
  blocks := [];
  list := AsList(H);

  for k in [1..Size(N.group_generators)] do
    g := N.group_generators[k];
    for i in [1..Size(S)] do
      for j in [1..Size(S)] do
        h := S[i]*g*S[j];
        if h in list then
          m := KGActionMatrix(M,[h]);
          blocks[i] := [i,j,m[1]];
          break;
        fi;
      od;
      mat := BlockMatrix(blocks,Size(S),Size(S));
      N.generators[k] := MatrixByBlockMatrix(mat);
    od;
  od;

  MakeImmutable(N.dimension);
  MakeImmutable(N.generators);

  return N;
end;

KGInducedModule := function(M,G)
  # Assert M.group is a subgroup of G
  if not IsSubgroup(G,M.group) then
    Error("KGInducedModule: M.group is not a subgroup of G");
  fi;
  return KGInducedModuleNC(M,G);
end;
```

4.3. Descomposición de KG-módulos

Para encontrar la descomposición del KG-módulo M como suma de inescindibles queremos encontrar primero una descomposición como suma de idempotentes ortogonales primitivos de la función identidad en M, id_M, que es el uno del álgebra $\mathrm{End}_{KG}(M)$, ya que existe una correspondencia entre los idempotentes de $\mathrm{End}_{KG}(M)$ (que son las proyecciones) y los sumandos directos de M, donde los idempotentes primitivos se corresponden con sumandos directos inescindibles de M. Luego, para obtener la descomposición de M, simplemente tomo los submódulos correspondientes a cada idempotente primitivo de la descomposición de id_M.

La estructura general de la parte encargada de descomponer los KG-módulos como suma de inescindibles es la siguiente:

⟨*Decomposing KG-modules*⟩≡
```
    # Decomposing KG-modules
    MAXITER := 2^17;
    MAXENDOSIZE := 2*MAXITER;
```
⟨*IsIdempotentMatrix*⟩
⟨*AreConjugatedMatrices*⟩
⟨*KGEndomorphismRing*⟩
⟨*ExternalDirectSummand*⟩
⟨*OrthogonalPrimitiveIdempotents*⟩
⟨*KGDecomposeModule*⟩

El algoritmo de descomposición

3. Proposición. Sea $\pi \in \mathrm{End}_{KG}(M)$. Entonces $M \cong \mathrm{Ker}(\pi) \oplus \mathrm{Im}(\pi)$.

Demostración: Como $m = \pi(m) + (m - \pi(m))$ y $\pi(m - \pi(m)) = 0$, tenemos que $M = \mathrm{Im}(\pi) + \mathrm{Ker}(\pi)$. Para ver que la suma es directa, sea $m = \pi(m_0) \in \mathrm{Ker}(\pi) \cap \mathrm{Im}(\pi)$. Entonces $0 = \pi(m) = \pi(\pi(m)) = m$, así que $\mathrm{Ker}(\pi) \cap \mathrm{Im}(\pi) = \{0\}$ ∎

4. Definición. Sea $\pi \in \mathrm{End}_{KG}(M)$ un idempotente. Decimos que el sumando directo de M que se corresponde con π es $\mathrm{Im}(\pi)$.

El siguiente algoritmo obtiene primero una descomposición de id_M como suma de idempotentes ortogonales primitivos, en seguida factoriza módulos isomorfos y, finalmente, devuelve una lista sin repeticiones (hasta isomorfismo) de los sumandos directos inescindibles de M.

⟨*KGDecomposeModule*⟩≡

```
# KGDecomposeModule
#
# Given a KG-module M returns a list S of idecomposable KG-modules
# such that M = \oplus_{s \in S} s
#
# Version 0.1
# Last modified September 27th, 2004
# Acme

KGDecomposeModule := function(M)
  local Endo, Idemp, OPI, S, c, d, Factor, NewFactor,
  CollectedFactors, U, f;

  Print("Generating the endomorphisms ring\n");
  Endo := KGEndomorphismRing( M );
  Hay("There are ",Size(Endo)," endomorphisms.\n");

  if Size(Endo)>MAXENDOSIZE then
    Print("The endomorphisms ring is too big to be kept in ",
          "memory. Aborting because swapping from memory to ",
          "disk would take too long.\n");
    return fail;
  fi;
  Print("Filtering idempotents\n");
  Idemp := IdempotentMatrices( Endo );

  Print("Removing zero from idempotents list\n");
  Idemp := Filtered(Idemp, x -> x <> Zero(x));
  Print("There are ",Size(Idemp)," idempotents left.\n");

  if Size(Idemp) > 1 then
    Print("Calculating a decomposition in primitive orthogonal ",
          "idempotents\n");
    OPI := OrthogonalPrimitiveIdempotents( Idemp );
    Print("Decomposition has ", Size(OPI), "summands. \n");

    Print("Collecting conjugated idempotents\n");
```

```
    # Collect isomorphic factors
    CollectedFactors := [];
    U := Units(Endo);
    Print("There are ", Size(U), "automorphisms. \n");
    for Factor in OPI do
      NewFactor := true;
      for c in [1..Size(CollectedFactors)] do
        f := CollectedFactors[c][1];
        if AreConjugatedMatrices(U, Factor, f) then
          NewFactor := false;
          CollectedFactors[c][2] := CollectedFactors[c][2] + 1;
          Print(c, "Old factor.\n");
          break;
        fi;
      od;
      if NewFactor then
        Add(CollectedFactors, [Factor,1]);
        Print("New factor.\n");
      fi;
    od;

    Print("Results of Decomposing this module: \n");
    Print(M, "\n");

    S := [];
    for c in [1..Size(CollectedFactors)] do
      S[c] := ExternalDirectSummand( M, CollectedFactors[c][1] );
      Print("Primitive Idempotent #", c, ". Repeated ",
            CollectedFactors[c][2]," times: \n");
      PrintArray(CollectedFactors[c][1]);
      Print("External Direct Summand #", c, ": \n");
      Print(S[c], "\n");
      if not MTX.IsIrreducible(S[c]) then
        Print("This summand is an idecomposable, but not ",
              "irreducible module.\n");
      fi;
    od;
    return S;
  else
    Print("Returning the trivial module.\n");
    return [KGTrivialModule(M.field, M.group)];
  fi;
end;
```

Calculando el anillo de endomorfismos

`KGEndomorphismRing` construye $\mathrm{End}_{KG}(M)$ utilizando las siguientes observaciones:

5. Observación.

$$\mathrm{End}_{KG}(M) = (\mathrm{End}_K(M))^G$$

Demostración:

$\subseteq$ Sea $\varphi \in \mathrm{End}_{KG}(M)$. $(g \star \varphi)(m) = g\varphi(g^{-1}m) = gg^{-1}\varphi(m) = \varphi(m)$.

$\supseteq$ Sea $\psi \in (\mathrm{End}_K(M))^G$. $\psi(m) = (g^{-1} \star \psi)(m) = g^{-1}\psi(gm)$ y, por lo tanto, $g\psi(m) = \psi(gm)$.

∎

6. Observación. Si M es un K espacio vectorial, entonces $\mathrm{End}_K(M) \cong GL_n(K)$, donde n es la dimensión de M.

7. Observación. Para que un K-endomomorfismo φ de M sea un KG-homomorfismo, basta que para toda $m \in M$ y para toda $g \in G$ $\varphi(gm) = g\varphi(m)$.

En este caso, φ tiene una representación matricial A, donde $A \in \mathrm{GL}_n(K)$ y $G \cong \mathrm{Aut}_{KG}(M)$ así que los elementos g de G son matrices de rango n, por lo tanto, pedir que $\varphi(gm) = g\varphi(m)$ es pedir que las matrices A y g conmuten.

⟨*KGEndomorphismRing*⟩≡

```
KGEndomorphismRing := function( M )
  local alg, sub, endo;

  alg := MatAlgebra(M.field, M.dimension);
  sub := Subalgebra(alg, M.generators);
  endo := Centralizer(alg, sub);

  return endo;
end;
```

Descomposición en Idempotentes Ortogonales Primitivos

Dada la lista de todos los idempotentes del anillo $\mathrm{End}_{KG}(M)$, quiero obtener una descomposición de la función identidad id_M como una suma de idempotentes ortogonales primitivos. Ya que estos se corresponden con sumandos directos del KG-módulo M.

Sean I_0 un idempotente fijo y M_0 el correspondiente submódulo de M. La idea básica del algoritmo es dividir la lista de idempotentes en tres listas: La primera, `Same`, debe contener a los idempotentes que se corresponden con sumandos directos de M_0; la segunda, `Ortho`, debe contener a los idempotentes que se corresponden con sumandos directos de $M_0^\perp$ (ie. del complemento ortogonal de M_0); y la última, `Other` debe contener a los idempotentes que se corresponden con sumandos directos de M que no están totalmente contenidos en M_0 ni totalmente contenidos en $M_0^\perp$, es decir, si denotamos por M_I al submódulo de M que se corresponde con I, los elementos de `Other` son los I tales que $M_I \cap M_0 \neq 0$ y $M_I \cap M_0^\perp \neq 0$.

El algoritmo es como sigue:

1. Si $I = 0$, entonces M_I es el subespacio trivial de M y no me sirve. Lo mando a `Other`;
2. Si $I_0 I = I I_0 = 0$, entonces M_0 y M_I son ortogonales entre sí, así que pongo a I en `Ortho`;
3. Si $I_0 I = I I_0 = I$, entonces M_I está contenido en M_0, así que I va a dar a `Same`;
4. En otro caso tenemos que $M_I \cap M_0 \neq 0$ y $M_I \cap M_0^\perp \neq 0$ y, por lo tanto, I va a dar a `Other`.

⟨ *OrthogonalPrimitiveIdempotents.Orthogonal/Same/Other* ⟩≡

```
if ( I = Zero(I_0) ) then
  NumOther := NumOther + 1;
  Other[NumOther] := I;
elif ( I*I_0 = Zero(I_0) ) and ( I_0*I = Zero(I_0) ) then
  NumOrtho := NumOrtho + 1;
  Ortho[NumOrtho] := I;
elif ( I*I_0 = I ) and ( I_0*I = I ) then
  NumSame := NumSame + 1;
  Same[NumSame] := I;
else
  NumOther := NumOther + 1;
  Other[NumOther] := I;
fi;
```

Note que, al final, `Same` contiene todos los idempotentes que se corresponden con todos los sumandos directos de M_0 y los elementos de `Ortho` se corresponden con todos los posibles sumandos directos de $M_0^{\perp}$, por lo tanto puedo llamar recursivamente a esta misma función para descomponer M_0 y $M_0^{\perp}$. El proceso termina porque los módulos tienen dimensión finita.

Por lo tanto cuando la función termine habremos partido a id_M en suma de idempotentes ortogonales primitivos. Sin importar cual haya sido el I_0 inicial, debemos llegar a la misma descomposición (hasta conjugación) pues el teorema de Krull-Smith-Azumaya nos dice que la descomposición de M en suma directa de inescindibles es esencialmente única. Vea la página 10.

⟨ *OrthogonalPrimitiveIdempotents*⟩≡

```
# OrthogonalPrimitiveIdempotents
#
# Given a list of idempotents (maybe including the zero), returns a list
# of orthogonal (the product is zero pairwise) primitive (their sum is the
# identity) idempotents.
#
# Version 0.2
# July 19th, 2004.
# Acme

OrthogonalPrimitiveIdempotents := function( IdempList )
  local c, I_0, I, Ortho, Same, Other, NumOrtho, NumSame, NumOther, OPI;

  if ( Size(IdempList) < 2 ) then
    if ( IdempList[1] = One(IdempList[1]) ) then
      return IdempList;
    else
      Error("OrthogonalPrimitiveIdempotents: list has only one ",
            "idempotent and it is NOT the identity");
    fi;
  fi;

  NumOrtho := 0;
  NumSame := 0;
  NumOther := 0;
  Ortho := [];
  Same := [];
  Other := [];

  I_0 := IdempList[1];
  for c in [2..Size(IdempList)] do
    I := IdempList[c];

    ⟨ OrthogonalPrimitiveIdempotents.Orthogonal/Same/Other ⟩
  od;
  NumSame := NumSame + 1;
  Same[NumSame] := I_0;

  OPI := [];
  if ( Size(Same)>1 ) then
```

```
      Append( OPI, OrthogonalPrimitiveIdempotents( Same ) );
   elif ( Size(Same)=1 ) then
      Add( OPI, Same[1] );
   fi;
   if ( Size(Ortho)>1 ) then
     Append( OPI, OrthogonalPrimitiveIdempotents( Ortho ) );
   elif ( Size(Ortho)=1 ) then
     Add( OPI, Ortho[1] );
   fi;

   return OPI;

   #A last-minute check
   if ( Sum(OPI) <> One(I_0) ) then
     Error("OrthogonalPrimitiveIdempotents: Output is NOT primitive as desired.\n");
   fi;
 end;
```

En el proceso de descomponer un módulo como suma directa de inescindibles, la mayor cantidad de tiempo de procesador (alrededor del 80 %) se gastaba en determinar cuales elementos del anillo de endomorfismos eran idempotentes.

El problema radicaba en que el algoritmo que usa GAP para decidir si x es idempotente es el siguiente: Calcula x^2; si $x^2 = x$, entonces x es idempotente, si no, pues no.

En nuestro caso, estamos interesados en saber si una matriz A es idempotente o no y, para concluir positivamente que A es idempotente, tenemos que calcular A^2. Pero para concluir que A no es idempotente nos basta con que una sola entrada de A^2 sea diferente a la correspondiente entrada de A.

El siguiente par de rutinas fueron escritas para evitar usar el algoritmo genérico de GAP , `IsIdempotentMatrix` decide si una matriz es idempotente o no; y `AreIdempotentMatrices` recibe una lista de matrices y devuelve solamente a las matrices idempotentes.

⟨*IsIdempotentMatrix*⟩≡

```
# IsIdempotentMatrix
#
# Returns true if the given squared matrix A is idempotent
# and false otherwise.
#
# Version 0.1
# Last modified September 27th, 2004
# Acme
IsIdempotentMatrix := function (A)
  local c;

  for c in [1..Size(A)] do
    if not (A[c]*A = A[c]) then
      return false;
    fi;
  od;

  return true;
end;

# AreIdempotentMatrices
#
# Recieves a list of squared matrices and returns the
# list of those which are idempotent.
#
# Version 0.1
# Last modified September 27th, 2004
# Acme
IdempotentMatrices := function( Matrices )
  local A, Idemp;

  Idemp := [];
  for A in AsList(Matrices) do
    if IsIdempotentMatrix(A) then
      Add(Idemp, A);
    fi;
  od;
```

```
  return Idemp;
end;
```

4.4. Identificando módulos isomorfos

Como para nuestros propósitos en realidad no ocupamos la descomposición de M como suma directa de inescindibles, lo que nos interesa es tener una lista sin repeticiones (hasta isomorfismo) de los submódulos inescindibles que aparecen en dicha descomposición. Así que necesitamos identificar módulos isomorfos.

Esto lo hacemos con la siguiente observación:

8. Observación. Sean $\pi_1, \pi_2 \in \mathrm{End}_{KG}(M)$ idempotentes. Entonces existe $\varphi \in \mathrm{Aut}_{KG}(M)$ tal que $\pi_1 = \varphi^{-1} \circ \pi_2 \circ \varphi$, si y sólo si los submódulos de M correspondientes a las proyecciones π_1 y π_2 son isomorfos.

Demostración:

$\Rightarrow$ Supongamos que $\pi_1 = \varphi^{-1} \circ \pi_2 \circ \varphi$. Sean $M_1 = \mathrm{Im}(\pi_1)$, $M_2 = \mathrm{Im}(\pi_2)$, y $x \in M_1$. Entonces existe $m \in M$ tal que $x = \pi_1(m) = (\varphi^{-1} \circ \pi_2 \circ \varphi)(m)$. Aplicando φ en ambos lados tenemos $\varphi(x) = (\pi_2 \circ \varphi)(m) \in M_2$. Por lo tanto la restricción de φ a M_1 va de M_1 a M_2 (en símbolos $\varphi|_{M_1} : M_1 \to M_2$) y es inyectiva porque φ lo era; y es suprayectiva porque $\mathrm{Ker}(\varphi|_{M_1}) \subset \mathrm{Ker}(\varphi) = \{0\}$ pues φ es isomorfismo.

$\Leftarrow$ Supongamos que $\psi : \mathrm{Im}(\pi_1) \to \mathrm{Im}(\pi_2)$ es un isomorfismo. Por la proposición 3, $M = \mathrm{Ker}(\pi_1) \oplus \mathrm{Im}(\pi_1) = \mathrm{Ker}(\pi_2) \oplus \mathrm{Im}(\pi_2)$, así que por el teorema de Krull-Smith-Azumaya (página 10), existe un isomorfismo $\bar{\psi} : \mathrm{Ker}(\pi_1) \to \mathrm{Ker}(\pi_2)$. Sea $\varphi : M = \mathrm{Ker}(\pi_1) \oplus \mathrm{Im}(\pi_1) \to M = \mathrm{Ker}(\pi_2) \oplus \mathrm{Im}(\pi_2)$ el homomorfismo que manda a $m = m_0 + m_1$ en $\bar{\psi}(m_0) + \psi(m_1)$. Entonces $(\varphi^{-1} \circ \pi_2 \circ \varphi)(m) = (\varphi^{-1} \circ \pi_2)(\bar{\psi}(m_0) + \psi(m_1)) = \varphi^{-1}(0 + \psi(m_1)) = m_1 = \pi_1(m)$.

∎

En el caso particular que nos interesa, $\mathrm{End}_{KG}(M)$ es isomorfo a un subgrupo de $\mathrm{GL}_n(K)$, donde n es la dimensión de M visto como K espacio vectorial. Así que nos basta identificar si dos matrices son o no conjugadas.

`AreConjugatedMatrices` nos ayuda a decidir si dadas dos matrices $A, B \in \mathrm{GL}_n(K)$ son conjugadas en un grupo $G \leq \mathrm{GL}_n(K)$. Esto es, si $A = gBg^{-1}$ para alguna $g \in G$. Esto es cierto si y sólo si $Ag = gB$ y g es invertible. Si G es grupo, entonces g es invertible, así que basta ver que $Ag = gB$. Al igual que en `IsIdempotentMatrix` existe un método en corto para decidir que no son conjugadas.

⟨*AreConjugatedMatrices*⟩≡

```
# AreConjugatedMatrices
#
# Last modified September 27th, 2004.
# Acme

AreConjugatedMatrices := function(G, A, B)
  local g, c, n;

  #if not (Size(A) = Size(B)) then
  #  return Error("Matrices A and B have different sizes");
  #elif not (Size(A) = Size(One(G))) then
  #  return Error("Matrices in G have wrong dimensions");
  #fi;
```

```
  n := Size(A);
  for g in G do
    # Calculate the c-th row of Ag and gB
    for c in [1..n] do
      if not (A[c]*g = g[c]*B) then
        break;
      elif c = n then
        return true;
      fi;
    od;
  od;

  return false;
end;
```

Recuperando los submódulos inescindibles

Como dijimos al principio de esta sección, existe una correspondencia entre los idempotentes primitivos del anillo de endomorfismos del KG-módulo M y los submódulos inescindibles de M. `ExternalDirectSummand` lleva los idempotentes a su correspondiente submódulo.

⟨*ExternalDirectSummand*⟩≡

```
# ExternalDirectSummand
#
# Given an idempotent non-zero matrix and a KG-module
# KG, ExternalDirectSummand returns a submodule M < KG
# such that the matrix is the natural projection from
# KG to M.
#
# Version 0.1
# Last modified September 27th, 2004
# Acme

ExternalDirectSummand := function( KG, gens )
  local V, M, basis, c, R, A;

  V := VectorSpace( KG.field, gens );
  basis := Basis( V );

  M := KGEmptyModule(KG.field,KG.group);
  M.dimension := Size(basis);

  for R in KG.generators do
    A := [];
    for c in [1..Size(basis)] do
      Add(A, Coefficients(basis, basis[c]*R));
    od;
    Add(M.generators,A);
  od;

  MakeImmutable(M.dimension);
  MakeImmutable(M.generators);

  return M;
end;
```

4.5. Módulos Simples

En esta sección tratamos con los algoritmos para decidir si un módulo es simple y para contar el número de KG-módulos simples. Aquí sólo tenemos dos componentes:

⟨*Meat Axe and simple KG-modules*⟩≡
```
# Simple modules and Meat Axe-dependent functions
⟨KGIsIrreducible⟩
⟨KGNumberOfIrreducibleModules⟩
```

El algoritmo que usamos para decidir si el módulo es simple es en realidad la implementación de Meat Axe que viene en GAP , de hecho las funciones `KGIsIrreducible` y `KGIsAbsolutelyIrreducible` son solamente sinónimos de las correspondientes funciones de Meat Axe . Vea los artículos de [Parker, 1984], [Holt and Rees, 1994] e [Ivanyos and Lux, 2000].

⟨*KGIsIrreducible*⟩≡
```
# KGIsIrreducible
#
# Irreducibility test for KG-modules using the MeatAxe
#
# Version 0.1
# June 28th, 2004
# Acme

KGIsIrreducible := MTX.IsIrreducible;
KGIsSimple := KGIsIrreducible;
KGIsAbsolutelyIrreducible := MTX.IsAbsolutelyIrreducible;
```

Sabemos que todo R-módulo tiene una serie de composición si y sólo si es Noetheriano y Artiniano ([Ash, 2000] 7.5.12)

Parece que según ([Alperin, 1986] p.1), si A es un álgebra de dimensión finita, todo A-módulo de dimensión finita tiene serie de composición. y todo KG-módulo simple es isomorfo a uno de los factores de composición (Luis).

Aquí es donde aprovechamos que nuestras estructuras para KG-módulos son compatibles con las de Meat Axe . Utilizamos Meat Axe primero para encontrar los factores de composición y luego para agruparlos en clases de isomorfismo, pues el número de estas clases de isomorfismo es el número de KG-módulos simples. Finalmente, usamos otra vez Meat Axe para extraer de la lista de KG-módulos simples a los que son absolutamente irreducibles.

⟨*KGNumberOfIrreducibleModules*⟩≡

```
# KGNumberOfIrreducibleModules
#
# Returns a list [ni, nai] where ni is the number of
# irreducible KG-modules and nai is the number of absolutely
# irreducible KG-modules.
#
# Version 0.1
# Last modified September 13th, 2004.
# Acme

KGNumberOfIrreducibleModules := function(K,G)
  local module, factors, f, m, numabsirr;

  module := KGRegularModule(K,G);
  factors := MTX.CollectedFactors(module);

  numabsirr := 0;
  for f in factors do
    m := f[1];
    if MTX.IsAbsolutelyIrreducible(m) then
      numabsirr := numabsirr+1;
    fi;
  od;

  return [Length(factors),numabsirr];
end;
```

4.6. Correspondientes de Green

En esta sección vamos a programar las construcciones que se hicieron en el capítulo 2.

⟨*Green correspondence*⟩≡
```
# Green Correspondence
⟨KGIsRelativelyProjective⟩
⟨KGIsProjective⟩
⟨KGVertex⟩
⟨KGGreenCorrespondentDownwards⟩
⟨KGGreenCorrespondentUpwards⟩
```

Los correspondientes de Green son calculados usando las siguientes rutinas:

⟨*KGGreenCorrespondentUpwards*⟩≡
```
# KGGreenCorrespondentUpwards
#
# Given a KH-module M returns an idecomposable KG-module N where
# N is a direct summand of M\uparrow_H^G and its vertex is G-conjugated to D.
# The argument D is a vertex of M
#
# Version 0.1
# June 28th, 2004
# Acme
KGGreenCorrespondentUpwardsNC := function(M,G,D)
  local MInd, S, N, V;

  MInd := KGInducedModule(M,G);
  S := KGDecomposeModule(MInd);

  for N in S do
    V := KGVertex(N);
    if IsConjugate(G, D, V) then
      return N;
    fi;
  od;
end;

KGGreenCorrespondentUpwards := function(M,G,D)
  # TODO: Assert H contains N_G(D)
  return KGGreenCorrespondentUpwardsNC(M,G,D);
end;
```

La siguiente rutina nunca se usa, está sólo por completez.

⟨*KGGreenCorrespondentDownwards*⟩≡

```
# KGGreenCorrespondentDownwards
#
# Given a KG-module M with vertex D, returns a KH-module N such that
# N is a direct summand of M\downarrow^G_H and its vertex V is H-conjugated to D.
#
# Version 0.1
# June 28, 2004
# Acme

KGGreenCorrespondentDownwardsNC := function(M, H, D)
  local MRest, S, N, V;

  MRest := KGRestrictedModule(M,H);
  S := KGDecomposeModule(MRest);

  for N in S do
    V := KGVertex(N);
    if IsConjugate(H, D, V) then
      return N;
    fi;
  od;

  return MRest;
end;

KGGreenCorrespondentDownwards := function(M, H, D)
  # TODO: Assert M es inescindible
  # TODO: Assert M es proyectivo relativo c.r. al p-subgrupo Q
  # TODO: Hacer que el parametro D sea opcional
  # TODO: Calcular el vertice D si no fue dado
  # TODO: Si D fue dado Assert D es un vertice de M

  return KGGreenCorrespondentDownwardsNC(M, H, D);
end;
```

En 21, definimos el vértice V del KG-módulo M como un subgrupo de G mínimo en el sentido de la inclusión, tal que M sea proyectivo relativo con respecto a V. Por 24 sabemos que V debe ser un p-grupo, para $p = \mathrm{Char}(K)$.

La rutina `KGVertex` busca un vértice de M generando la lattice de p-subgrupos de G y recorriéndola comenzando con el subgrupo trivial y ascendiendo hacia G.

⟨*KGVertex*⟩≡

```
# KGVertex
# Calculate the Vertex of a KG-module M.
#
# Version 0.1
# June 11th, 2004.
# Acme
KGVertex := function( M )
  local D, lat, subgroups, p, tmp;

  Print("Computing the vertex...\n");

  tmp := KG.IsProjective(M);
  if tmp = fail then
    return fail;
  fi;

  if tmp then
    return TrivialSubgroup(M.group);
  else
    Print("Computing subgroups lattice...\n");
    lat := LatticeSubgroups( M.group );

    Print("Finding p-groups...\n");
    subgroups := List( ConjugacyClassesSubgroups(lat), Representative );
    p := Characteristic(M.field);
    # Next line will delete the trivial subgroup from the list among others
    subgroups := Filtered( subgroups, x -> (Size(x) mod p = 0) and IsPGroup(x) );
    Sort( subgroups, function(g,h) return Size(g) < Size(h); end );

    Print("KGVertex: Trying to find a vertex in: ");
    for D in subgroups do
      Print(IdGroup(D), ", ");
      tmp := KGIsRelativelyProjective(M, D, MAXITER);
      if tmp = true then
        Print("... found!\n");
        return D;
      elif tmp = fail then
        Print("... failed!\n");
        return fail;
      fi;
    od;

    return M.group;
  fi;
end;
```

Alperin [Alperin, 1986] (p. 33) nos da un criterio rápido para decidir si un módulo NO es proyectivo. La notación del libro implica que p es la característica del campo K.

9. Teorema. Si un p-subgrupo de Sylow P tiene orden p^a, entonces todo KG-módulo proyectivo tiene dimensión divisible por p^a

Utilizamos este criterio para descartar rápidamente algunos módulos que no pueden ser proyectivos, y luego nos preguntamos si el módulo es proyectivo relativo con respecto al subgrupo trivial.

⟨*KGIsProjective*⟩≡

```
KG.IsProjective := KG.Attribute("isProjective",
  function(M)
    local q, tmp, E;

    q := Size( SylowSubgroup(M.group,Characteristic(M.field)) );
    if not (M.dimension mod q = 0) then
      M.isProjective := false;
    else
      E := TrivialSubgroup(M.group);
      M.isProjective := KGIsRelativelyProjective(M,E,MAXITER);
    fi;

    return KG.IsProjective(M);
  end
);
```

El criterio de Higman (vea 18) nos dice que un KG-módulo M es proyectivo relativo con respecto a H si y sólo si la función identidad en M está en la imagen de la traza relativa $Tr_{H,G}$. La traza relativa se definió en la página 19.

Actualmente esta es la rutina que se lleva con mucho el mayor tiempo de procesador. Para poder usarla con grupos de ordenes superiores a 150 es mejor dedicarse un rato a pensar como cambiarla por un criterio más eficiente.

`KGIsRelativelyProjective` calcula uno a uno los elementos de $\mathrm{Im}(\mathrm{Tr}_{H,G})$, si encuentra a id_M puede concluir positivamente que el módulo es proyectivo relativo con respecto al parámetro H. Pero para poder concluir que el módulo no es proyectivo relativo con respecto a H debe construir toda la imagen de $\mathrm{Tr}_{H,G}$. Y cuando uno anda buscando un vértice normalmente se obtienen varios no antes de obtener un sí.

⟨*KGIsRelativelyProjective*⟩≡

```
# KGIsRelativelyProjective
# Decide if a KG-módule M is relatively H projective
# that happens iff Tr_{H,G} is suprajective
#
# Version 0.1
# Last modified September 17th, 2004.
# Acme

KGIsRelativelyProjective := function(KG,H,MAXITER)
  local KH, source, cosets, A, B, c, F, S, s, iter;

  KH := KGRestrictedModule( KG, H );

  source := KGEndomorphismRing( KH );

  if Size(source) > 10^4 then
    Print("KGIsRelativelyProjective: Take a seat and make coffee, this is gonna take
    Print("The endomorphism ring has ", Size(source), " squared matrices of size ", K
    if Size(source) > 10^6 then
      Print("There are few chances of success, better call this off at once\n");
      return fail;
    fi;
  fi;

  cosets := RightTransversal(KG.group,H);
  A := KGActionMatrix( KG, cosets );

  B := [];
  for c in [1..Size(A)] do
    B[c] := Inverse(A[c]);
  od;

  iter := 0;
  for F in source do
    S := [];
    for c in [1..Size(A)] do
      S[c] := B[c]*F*A[c];
    od;
    s := Sum(S);
```

```
      if ( s = One(s) ) then
        return true;
        break;
      fi;
      iter := iter + 1;
      if iter mod 10000 = 0 then
        Print(iter," elements tested so far (", QuoInt(100*iter,Size(source)), "%)\n");
        if iter > MAXITER then
          #Print("MAXITER reached, press return to continue");
          return fail;
        fi;
      fi;
    od;

    return false;
  end;
```

4.7. Conjetura de Alperin y sus posibles variantes

Como dijimos en el capítulo 3, la conjetura de Alperin es que, si K es un campo algebraicamente cerrado, el número de KG-módulos simples hasta isomorfismo es igual al número de pesos para KG hasta conjugación.

Y, en ese caso, el número de KG-módulos simples es igual al número de KG-módulos absolutamente irreducibles y al número de clases de conjugación p-regulares de G.

Pero ningún campo finito es algebraicamente cerrado y, en este caso, normalmente las tres cantidades mencionadas difieren. Entonces nos hemos fijado como tarea responder a las siguientes cuestiones: ¿Cómo se debería enunciar la conjetura de Alperin?, ¿sigue siendo una conjetura razonable aún para campos que no son algebraicamente cerrados?, ¿podemos proponer alguna otra conjetura?.

La estructura de esta parte del código es:

⟨*Alperin conjecture*⟩≡

```
    # Alperin's conjecture and possible alternative conjectures
    ⟨KGWeights⟩
    ⟨CountPRegularConjugacyClasses⟩
    ⟨KGCheckConjeture⟩
```

Pesos

En su artículo [Alperin, 1987], Alperin define un **peso** para la pareja (K, G) como una pareja (Q, S) en donde

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

`KGWeights` genera una lista de pesos para KG. Para calcular los pesos, esta rutina recibe un campo K, cuya característica p divida al orden de G. Luego calcula el módulo de permutación M, definido como $K\left[\frac{G}{P}\right]$, donde P es un p-subgrupo de Sylow de G, luego descompone a M como suma de módulos inescindibles y, a cada sumando inescindible N de M le calcula su correspondiente de Green S, si S es simple, entonces tiene un peso (Q, S).

⟨*KGWeights*⟩≡

```
# KGWeights
#
# Find all the weights (Q,S) for a pair (K,G)
#
# Version 0.1
# July 20th, 2004
# Acme

KGWeights := function(K, G)
  local nw, weights, P, M, MS, N, Q, S;

  P := SylowSubgroup(G, Characteristic(K));
  M := KGPermutationModule(K,G,P);
  MS := KGDecomposeModule(M);
  if MS = fail then
    return fail;
  fi;

  nw := 0;
  weights := [];
  for N in MS do
    Q := KGVertex(N);
    if Q = fail then
      return fail;
    fi;
    S := KGGreenCorrespondentDownwards(N,Normalizer(G,Q),Q);
    Print("Checking if the module is Simple...\n");
    if KGIsIrreducible(S) then
      nw := nw + 1;
      weights[nw] := [Q,S];
    fi;
  od;

  return weights;
end;
```

Clases de Conjugación p-Regulares

10. Definición. Sean G un grupo, $g_0 \in G$, y p un primo. La **clase de conjugación** de g_0 es la órbita de g_0 bajo la acción de conjugación, esto es $\{gg_0g^{-1} | g \in G\}$.

Sabemos (ver siguiente párrafo) que todos los elementos de una clase de conjugación tienen el mismo orden, así que podemos decir que una clase de conjugación es **p-regular** si el orden de un elemento cualquiera de dicha clase de conjugación es primo relativo con p.

Sean $r, s, g \in G$ tales que $r = gsg^{-1}$, con $|r| = n$. Entonces $e = r^n = (gsg^{-1})^n = gs^ng^{-1}$ conjugando por g^{-1} tenemos $s = g^{-1}eg = e$. Por lo tanto el orden de s divide a n, pero una argumentación análoga usando que $s = g^{-1}rg$ nos dice que el orden de r divide al de s y por lo tanto son iguales.

`CountPRegularConjugacyClasses` Cuenta las clases de conjugación p-regulares encontrando primero todas las clases de conjugación de G y luego viendo cuales de ellas son p-regulares.

⟨ *CountPRegularConjugacyClasses* ⟩≡

```
# CountPRegularConjugacyClasses
#
# Given a group G and a prime p, returns the number of
# p-regular conjugacy classes. That is, the number of
# conjugacy classes whose elements' order is relatively prime
# to p.
#
# Version 0.1
# Last modified July 3th, 2004
# Last revision September 7th, 2004
# Acme
CountPRegularConjugacyClassesNC := function(G, p)
  local NumCC, CC, c, g;

  CC := ConjugacyClasses(G);

  NumCC := 0;
  for c in [1..Size(CC)] do
    g := Representative( CC[c] );
    if Gcd(Order(g),p) = 1 then
      NumCC := NumCC + 1;
    fi;
  od;

  return NumCC;
end;

CountPRegularConjugacyClasses := function(G, p)
  # Assert that G is group and p is prime, then call the NC version
  if not IsGroup(G) then
    Error("CountPRegularConjugacyClasses: Error, G is not a group");
    ReturnFail();
  elif not IsPrime(p) then
    Error("CountPRegularConjugacyClasses: Error, p is not prime");
```

```
    ReturnFail();
  else
    return CountPRegularConjugacyClassesNC(G, p);
  fi;
end;
```

La rutina principal

Sabiendo que, en general, las cardinalidades de

- el conjunto de clases de conjugación de G;
- el conjunto de clases de conjugación p-regulares de G;
- el conjunto de KG-módulos simples; y
- el conjunto de KG-módulos absolutamente irreducibles

no tienen por que ser iguales, hicimos una rutina que calcula el número de pesos de KG y los compara con los otros cuatro antes mencionados. Así que la mandamos llamar una vez para cada pareja (K, G) y filtramos la salida para obtener la tabla de resultados.

⟨*KGCheckConjeture*⟩≡

```
# KGCheckConjeture
#
# Version 0.1
# July 6th, 2004
# Acme

KGCheckConjeture := function(K,G)
  local Weights, NumCC, NumPRCC, NumIrr, NumAbsIrr, tmp, c, Result;

  Print("\n\n\n\nStarting a new problem...\n");
  Print("Field has ", Size(K), " elements\n");
  Print("Group ID is ", IdGroup(G), "\n");

  Weights := KGWeights(K, G);
  if Weights = fail then
    Print("#I Group ID is ", IdGroup(G), "\n");
    Print("#I Field has ", Size(K), " elements\n");
    Print("#I FAILED to compute weights!!!\n");
    return fail;
  fi;

  NumCC := Size(ConjugacyClasses(G));
  NumPRCC := CountPRegularConjugacyClasses(G,Characteristic(K));
  tmp := KGNumberOfIrreducibleModules(K,G);
  NumIrr := tmp[1];
  NumAbsIrr := tmp[2];

  for c in [1..20] do Print("########"); od;
  Print("\n#I So... the results are as follows: \n");
  Print("#I Group ID is ", IdGroup(G), "\n");
  Print("#I Field has ", Size(K), " elements\n");
  Print("#I There are: \n");
  Print("#I ", Size(Weights), " Weights; \n");
  Print("#I ", NumCC, " Conjugacy Classes; \n");
  Print("#I ", NumPRCC, " P-Regular Conjugacy Classes; \n");
  Print("#I ", NumIrr," Irreducible KG-modules; and \n");
```

```
  Print("#I ", NumAbsIrr, " Absolutely Irreducible KG-modules \n");
  Print("#I ", IdGroup(G), " & ",Size(Weights), " & ",NumCC, " & ",NumPRCC, " & ",Num
  if Size(Weights) = NumIrr then
    Print("Conjecture is Fullfilled!!!\n");
    Result := [true];
    if (NumPRCC = NumIrr) and (NumIrr = NumAbsIrr) then
      Add(Result, true);
    else
      Add(Result, false);
    fi;
  else
    Print("Conjecture Failed !!!!!!!!!!\n");
    Print("Weights follows:\n");
    for c in [1..Size(Weights)] do
      Print("Weight #",c,": \n",Weights[c],"\n\n");
    od;
    return [false, false];
  fi;
end;
```

Capítulo 5

Resultados y Conclusiones

Las tablas obtenidas a partir de los ejemplos corridos en GAP se muestran en seguida. El caso $\mathbb{F}_2\mathbb{Z}_6$ merece especial atención ya que es el ejemplo más pequeño en el cual dejan de coincidir el número de KG-módulos simples con el de KG-módulos absolutamente irreducibles y con el número de clases de conjugación p-regulares.

La notación es como sigue

K En esta columna aparece el orden del campo K;

G Aquí aparece el identificador de GAP para el grupo G;

W Denota el número de pesos de KG;

C Denota el número de clases de conjugación de G;

P Denota el número de clases de conjugación p-regulares de G, donde p es la característica de K;

S Denota el número de KG-módulos simples;

A Denota el número de KG-módulos absolutamente irreducibles.

K	G	W	C	P	S	A
2	[2, 1]	1	2	1	1	1
2	[4, 1]	1	4	1	1	1
2	[4, 2]	1	4	1	1	1
2	[6, 1]	2	3	2	2	2
2	[6, 2]	2	6	3	2	1
2	[8, 1]	1	8	1	1	1
2	[8, 2]	1	8	1	1	1
2	[8, 3]	1	5	1	1	1
2	[8, 4]	1	5	1	1	1
2	[8, 5]	1	8	1	1	1
2	[10, 1]	2	4	3	2	1
2	[10, 2]	2	10	5	2	1
2	[12, 1]	2	6	2	2	2
2	[12, 2]	2	12	3	2	1
2	[12, 3]	2	4	3	2	1

K	G	W	C	P	S	A
2	[12, 4]	2	6	2	2	2
2	[12, 5]	2	12	3	2	1
2	[14, 1]					
2	[14, 2]	3	14	7	3	1
2	[16, 1]	1	16	1	1	1
2	[16, 2]	1	16	1	1	1
2	[16, 3]	1	10	1	1	1
2	[16, 4]	1	10	1	1	1
2	[16, 5]	1	16	1	1	1
2	[16, 6]	1	10	1	1	1
2	[16, 7]	1	7	1	1	1
2	[16, 8]	1	7	1	1	1
2	[16, 9]	1	7	1	1	1
2	[16, 10]	1	16	1	1	1
2	[16, 11]	1	10	1	1	1
2	[16, 12]	1	10	1	1	1
2	[16, 13]	1	10	1	1	1
2	[16, 14]	1	16	1	1	1
2	[18, 1]					
2	[18, 2]					
2	[18, 3]	4	9	6	4	2
2	[18, 4]	5	6	5	5	5
2	[18, 5]	5	18	9	5	1
2	[20, 1]	2	8	3	2	1
2	[20, 2]	2	20	5	2	1
2	[20, 3]	2	5	2	2	2
2	[20, 4]	2	8	3	2	1
2	[20, 5]	2	20	5	2	1
2	[22, 1]					
2	[22, 2]					
2	[24, 1]	2	12	2	2	2
2	[24, 2]	2	24	3	2	1
2	[24, 3]	2	7	3	2	1
2	[24, 4]	2	9	2	2	2
2	[24, 5]	2	12	2	2	2
2	[24, 6]	2	9	2	2	2
2	[24, 7]	2	12	2	2	2
2	[24, 8]	2	9	2	2	2
2	[24, 9]	2	24	3	2	1
2	[24, 10]	2	15	3	2	1
2	[24, 11]	2	15	3	2	1
2	[24, 12]	2	5	2	2	2
2	[24, 13]	2	8	3	2	1
2	[24, 14]	2	12	2	2	2
2	[24, 15]	2	24	3	2	1
2	[26, 1]					
2	[26, 2]					
2	[28, 1]					
2	[28, 2]	3	28	7	3	1
2	[28, 3]					
2	[28, 4]	3	28	7	3	1

K	G	W	C	P	S	A
2	[30, 1]					
2	[30, 2]	5	12	9	5	1
2	[30, 3]					
2	[30, 4]	5	30	15	5	1
2	[32, 1]	1	32	1	1	1
2	[32, 2]	1	20	1	1	1
2	[32, 3]	1	32	1	1	1
2	[32, 4]	1	20	1	1	1
2	[32, 5]	1	20	1	1	1
2	[32, 6]	1	11	1	1	1
2	[32, 7]	1	11	1	1	1
2	[32, 8]	1	11	1	1	1
2	[32, 9]	1	14	1	1	1
2	[32, 10]	1	14	1	1	1
2	[32, 11]	1	14	1	1	1
2	[32, 12]	1	20	1	1	1
2	[32, 13]	1	14	1	1	1
2	[32, 14]	1	14	1	1	1
2	[32, 15]	1	14	1	1	1
2	[32, 16]	1	32	1	1	1
2	[32, 17]	1	20	1	1	1
2	[32, 18]	1	11	1	1	1
2	[32, 19]	1	11	1	1	1
2	[32, 20]	1	11	1	1	1
2	[32, 21]	1	32	1	1	1
2	[32, 22]	1	20	1	1	1
2	[32, 23]	1	20	1	1	1
2	[32, 24]	1	20	1	1	1
2	[32, 25]	1	20	1	1	1
2	[32, 26]	1	20	1	1	1
2	[32, 27]	1	14	1	1	1
2	[32, 28]	1	14	1	1	1
2	[32, 29]	1	14	1	1	1
2	[32, 30]	1	14	1	1	1
2	[32, 31]	1	14	1	1	1
2	[32, 32]	1	14	1	1	1
2	[32, 33]	1	14	1	1	1
2	[32, 34]	1	14	1	1	1
2	[32, 35]	1	14	1	1	1
2	[32, 36]	1	32	1	1	1
2	[32, 37]	1	20	1	1	1
2	[32, 38]	1	20	1	1	1
2	[32, 39]	1	14	1	1	1
2	[32, 40]	1	14	1	1	1
2	[32, 41]	1	14	1	1	1
2	[32, 42]	1	14	1	1	1
2	[32, 43]	1	11	1	1	1
2	[32, 44]	1	11	1	1	1
2	[32, 45]	1	32	1	1	1
2	[32, 46]	1	20	1	1	1
2	[32, 47]	1	20	1	1	1

K	G	W	C	P	S	A
2	[32, 48]	1	20	1	1	1
2	[32, 49]	1	17	1	1	1
2	[32, 50]	1	17	1	1	1
2	[32, 51]	1	32	1	1	1
2	[34, 1]					
2	[38, 1]					
2	[38, 2]					
2	[40, 1]	2	16	3	2	1
2	[40, 2]	2	40	5	2	1
2	[40, 3]	2	10	2	2	2
2	[40, 4]	2	13	3	2	1
2	[40, 5]	2	16	3	2	1
2	[40, 6]	2	13	3	2	1
2	[40, 7]	2	16	3	2	1
2	[40, 8]	2	13	3	2	1
2	[40, 9]	2	40	5	2	1
2	[40, 10]	2	25	5	2	1
2	[40, 11]	2	25	5	2	1
2	[40, 12]	2	10	2	2	2
2	[40, 13]	2	16	3	2	1
2	[40, 14]	2	40	5	2	1
2	[42, 1]					
2	[42, 2]					
2	[42, 3]					
2	[42, 4]					
2	[42, 5]					
2	[42, 6]					
2	[44, 1]					
2	[44, 2]					
2	[44, 3]					
2	[44, 4]					
2	[46, 1]					
2	[46, 2]					
2	[48, 1]	2	24	2	2	2
2	[48, 2]	2	48	3	2	1
2	[48, 3]	2	8	3	2	1
2	[48, 4]	2	24	2	2	2
2	[48, 5]	2	18	2	2	2
2	[48, 6]	2	15	2	2	2
2	[48, 7]	2	15	2	2	2
2	[48, 8]	2	15	2	2	2
2	[48, 9]	2	24	2	2	2
2	[48, 10]	2	18	2	2	2
2	[48, 11]	2	24	2	2	2
2	[48, 12]	2	18	2	2	2
2	[48, 13]	2	18	2	2	2
2	[48, 14]	2	18	2	2	2
2	[48, 15]	2	12	2	2	2
2	[48, 16]	2	12	2	2	2
2	[48, 17]	2	12	2	2	2
2	[48, 18]	2	12	2	2	2

K	G	W	C	P	S	A
2	[48, 19]	2	18	2	2	2
2	[48, 20]	2	48	3	2	1
2	[48, 21]	2	30	3	2	1
2	[48, 22]	2	30	3	2	1
2	[48, 23]	2	48	3	2	1
2	[48, 24]	2	30	3	2	1
2	[48, 25]	2	21	3	2	1
2	[48, 26]	2	21	3	2	1
2	[48, 27]	2	21	3	2	1
2	[48, 28]	2	8	2	2	2
2	[48, 29]	2	8	2	2	2
2	[48, 30]	2	10	2	2	2
2	[48, 31]	2	16	3	2	1
2	[48, 32]	2	14	3	2	1
2	[48, 33]	2	14	3	2	1
2	[48, 34]	2	18	2	2	2
2	[48, 35]	2	24	2	2	2
2	[48, 36]	2	18	2	2	2
2	[48, 37]	2	18	2	2	2
2	[48, 38]	2	15	2	2	2
2	[48, 39]	2	15	2	2	2
2	[48, 40]	2	15	2	2	2
2	[48, 41]	2	15	2	2	2
2	[48, 42]	2	24	2	2	2
2	[48, 43]	2	18	2	2	2
2	[48, 44]	2	48	3	2	1
2	[48, 45]	2	30	3	2	1
2	[48, 46]	2	30	3	2	1
2	[48, 47]	2	30	3	2	1
2	[48, 48]	2	10	2	2	2
2	[48, 49]	2	16	3	2	1
2	[48, 50]	2	8	3	2	1
2	[48, 51]	2	24	2	2	2
2	[48, 52]	2	48	3	2	1
2	[50, 1]					
2	[50, 2]					
2	[50, 3]					
2	[50, 4]	7	14	13	7	1
2	[50, 5]					
2	[52, 1]					
2	[52, 2]					
2	[52, 3]					
2	[52, 4]					
3	[3, 1]	1	3	1	1	1
3	[6, 1]	2	3	2	2	2
3	[6, 2]	2	6	2	2	2
3	[9, 1]	1	9	1	1	1
3	[9, 2]	1	9	1	1	1
3	[12, 1]	3	6	4	3	2
3	[12, 2]	3	12	4	3	2
3	[12, 3]	2	4	2	2	2

K	G	W	C	P	S	A
3	[12, 4]	4	6	4	4	4
3	[12, 5]	4	12	4	4	4
3	[15, 1]					
3	[18, 1]	2	6	2	2	2
3	[18, 2]	2	18	2	2	2
3	[18, 3]	2	9	2	2	2
3	[18, 4]	2	6	2	2	2
3	[18, 5]	2	18	2	2	2
3	[21, 1]					
3	[21, 2]					
3	[24, 1]	5	12	8	5	2
3	[24, 2]	5	24	8	5	2
3	[24, 3]	3	7	3	3	3
3	[24, 4]	5	9	5	5	5
3	[24, 5]	6	12	8	6	4
3	[24, 6]	5	9	5	5	5
3	[24, 7]	6	12	8	6	4
3	[24, 8]	5	9	5	5	5
3	[24, 9]	6	24	8	6	4
3	[24, 10]	5	15	5	5	5
3	[24, 11]	5	15	5	5	5
3	[24, 12]	4	5	4	4	4
3	[24, 13]	4	8	4	4	4
3	[24, 14]	8	12	8	8	8
3	[24, 15]	8	24	8	8	8
3	[27, 1]	1	27	1	1	1
3	[27, 2]	1	27	1	1	1
3	[27, 3]	1	11	1	1	1
3	[27, 4]	1	11	1	1	1
3	[27, 5]	1	27	1	1	1
3	[30, 1]					
3	[30, 2]					
3	[30, 3]					
3	[30, 4]					
3	[36, 1]	3	12	4	3	2
3	[36, 2]	3	36	4	3	2
3	[36, 3]	2	12	2	2	2
3	[36, 4]	4	12	4	4	4
3	[36, 5]	4	36	4	4	4
3	[36, 6]	3	18	4	3	2
3	[36, 7]	3	12	4	3	2
3	[36, 8]	3	36	4	3	2
3	[36, 9]	3	6	4	3	2
3	[36, 10]	4	9	4	4	4
3	[36, 11]	2	12	2	2	2
3	[36, 12]	4	18	4	4	4
3	[36, 13]	4	12	4	4	4
3	[36, 14]	4	36	4	4	4
3	[39, 1]	5	7	5	5	5
3	[39, 2]					
3	[42, 1]					

K	G	W	C	P	S	A
3	[42, 2]					
3	[42, 3]					
3	[42, 4]					
3	[42, 5]					
3	[42, 6]					
3	[45, 1]					
3	[45, 2]					
3	[48, 1]					
3	[48, 2]					
3	[48, 3]					
3	[48, 4]					
3	[48, 5]					
3	[48, 6]					
3	[48, 7]					
3	[48, 8]					
3	[48, 9]					
3	[48, 10]					
3	[48, 11]					
3	[48, 12]					
3	[48, 13]					
3	[48, 14]					
3	[48, 15]					
3	[48, 16]					
3	[48, 17]					
3	[48, 18]					
3	[48, 19]					
3	[48, 20]					
3	[48, 21]					
3	[48, 22]					
3	[48, 23]					
3	[48, 24]					
3	[48, 25]					
3	[48, 26]					
3	[48, 27]					
3	[48, 28]					
3	[48, 29]	6	8	6	6	6
3	[48, 30]					
3	[48, 31]					
3	[48, 32]	6	14	6	6	6
3	[48, 33]					
3	[48, 34]					
3	[48, 35]					
3	[48, 36]					
3	[48, 37]					
3	[48, 38]					
3	[48, 39]					
3	[48, 40]					
3	[48, 41]					
3	[48, 42]					
3	[48, 43]					
3	[48, 44]					

K	G	W	C	P	S	A
3	[48, 45]					
3	[48, 46]					
3	[48, 47]					
3	[48, 48]	8	10	8	8	8
3	[48, 49]	8	16	8	8	8
3	[48, 50]	6	8	6	6	6
3	[48, 51]					
3	[48, 52]					
3	[51, 1]					
3	[54, 1]	2	15	2	2	2
3	[54, 2]	2	54	2	2	2
3	[54, 3]	2	18	2	2	2
3	[54, 4]	2	27	2	2	2
3	[54, 5]	2	10	2	2	2
3	[54, 6]	2	10	2	2	2
3	[54, 7]	2	15	2	2	2
3	[54, 8]	2	10	2	2	2
3	[54, 9]	2	54	2	2	2
3	[54, 10]	2	22	2	2	2
3	[54, 11]	2	22	2	2	2
3	[54, 12]	2	27	2	2	2
3	[54, 13]	2	18	2	2	2
3	[54, 14]	2	15	2	2	2
3	[54, 15]	2	54	2	2	2
3	[57, 1]					
3	[57, 2]					
3	[60, 1]					
3	[60, 2]					
3	[60, 3]					
3	[60, 4]					
3	[60, 5]					
3	[60, 6]					
3	[60, 7]					
3	[60, 8]					
3	[60, 9]					
3	[60, 10]					
3	[60, 11]					
3	[60, 12]					
3	[60, 13]					
5	[5, 1]	1	5	1	1	1
5	[10, 1]	2	4	2	2	2
5	[10, 2]	2	10	2	2	2
5	[15, 1]	2	15	3	2	1
5	[20, 1]	4	8	4	4	4
5	[20, 2]	4	20	4	4	4
5	[20, 3]	4	5	4	4	4
5	[20, 4]	4	8	4	4	4
5	[20, 5]	4	20	4	4	4
5	[25, 1]	1	25	1	1	1
5	[25, 2]	1	25	1	1	1
5	[30, 1]	3	15	3	3	3

K	G	W	C	P	S	A
5	[30, 2]	4	12	6	4	2
5	[30, 3]	3	9	3	3	3
5	[30, 4]	4	30	6	4	2
5	[35, 1]					
5	[40, 1]					
5	[40, 2]					
5	[40, 3]					
5	[40, 4]					
5	[40, 5]					
5	[40, 6]					
5	[40, 7]					
5	[40, 8]					
5	[40, 9]					
5	[40, 10]					
5	[40, 11]					
5	[40, 12]					
5	[40, 13]					
5	[40, 14]					
5	[45, 1]					
5	[45, 2]					
5	[50, 1]	2	14	2	2	2
5	[50, 2]	2	50	2	2	2
5	[50, 3]	2	20	2	2	2
5	[50, 4]	2	14	2	2	2
5	[50, 5]	2	50	2	2	2
5	[55, 1]					
5	[55, 2]					
5	[60, 1]					
5	[60, 2]					
5	[60, 3]					
5	[60, 4]					
5	[60, 5]					
5	[60, 6]					
5	[60, 7]					
5	[60, 8]					
5	[60, 9]					
5	[60, 10]					
5	[60, 11]					
5	[60, 12]					
5	[60, 13]					
5	[65, 1]					
5	[70, 1]					
5	[70, 2]					
5	[70, 3]					
5	[70, 4]					
5	[75, 1]	2	75	3	2	1
5	[75, 2]	2	11	3	2	1
5	[75, 3]	2	75	3	2	1
5	[80, 1]					
5	[80, 2]					
5	[80, 3]					

K	G	W	C	P	S	A
5	[80, 4]					
5	[80, 5]					
5	[80, 6]					
5	[80, 7]					
5	[80, 8]					
5	[80, 9]					
5	[80, 10]					
5	[80, 11]					
5	[80, 12]					
5	[80, 13]					
5	[80, 14]					
5	[80, 15]					
5	[80, 16]					
5	[80, 17]					
5	[80, 18]					
5	[80, 19]					
5	[80, 20]					
5	[80, 21]					
5	[80, 22]					
5	[80, 23]					
5	[80, 24]					
5	[80, 25]					
5	[80, 26]					
5	[80, 27]					
5	[80, 28]					
5	[80, 29]					
5	[80, 30]					
5	[80, 31]					
5	[80, 32]					
5	[80, 33]					
5	[80, 34]					
5	[80, 35]					
5	[80, 36]					
5	[80, 37]					
5	[80, 38]					
5	[80, 39]					
5	[80, 40]					
5	[80, 41]					
5	[80, 42]					
5	[80, 43]					
5	[80, 44]					
5	[80, 45]					
5	[80, 46]					
5	[80, 47]					
5	[80, 48]					
5	[80, 49]					
5	[80, 50]					
5	[80, 51]					
5	[80, 52]					
5	[85, 1]					
5	[90, 1]					

K	G	W	C	P	S	A
5	[90, 2]					
5	[90, 3]					
5	[90, 4]					
5	[90, 5]					
5	[90, 6]					
5	[90, 7]					
5	[90, 8]					
5	[90, 9]					
5	[90, 10]					
5	[95, 1]					
5	[100, 1]	4	28	4	4	4
5	[100, 2]	4	100	4	4	4
5	[100, 3]	4	10	4	4	4
5	[100, 4]	4	28	4	4	4
5	[100, 5]	4	100	4	4	4
5	[100, 6]	4	40	4	4	4
5	[100, 7]	4	28	4	4	4
5	[100, 8]	4	100	4	4	4
5	[100, 9]	4	25	4	4	4
5	[100, 10]	4	13	4	4	4
5	[100, 11]	4	10	4	4	4
5	[100, 12]	4	10	4	4	4
5	[100, 13]	4	16	4	4	4
5	[100, 14]	4	40	4	4	4
5	[100, 15]	4	28	4	4	4
5	[100, 16]	4	100	4	4	4
5	[60, 1]					
5	[60, 2]					
5	[60, 3]					
5	[60, 4]					
5	[60, 5]					
5	[60, 6]					
5	[60, 7]					
5	[60, 8]					
5	[60, 9]					
5	[60, 10]					
5	[60, 11]					
5	[60, 12]					
5	[60, 13]					
5	[65, 1]					
5	[70, 1]					
5	[70, 2]					
5	[70, 3]					
5	[70, 4]					
5	[75, 1]	2	75	3	2	1
5	[75, 2]	2	11	3	2	1
5	[75, 3]	2	75	3	2	1
5	[80, 1]					
5	[80, 2]					
5	[80, 3]					
5	[80, 4]					

K	G	W	C	P	S	A
5	[80, 5]					
5	[80, 6]					
5	[80, 7]					
5	[80, 8]					
5	[80, 9]					
5	[80, 10]					
5	[80, 11]					
5	[80, 12]					
5	[80, 13]					
5	[80, 14]					
5	[80, 15]					
5	[80, 16]					
5	[80, 17]					
5	[80, 18]					
5	[80, 19]					
5	[80, 20]					
5	[80, 21]					
5	[80, 22]					
5	[80, 23]					
5	[80, 24]					
5	[80, 25]					
5	[80, 26]					
5	[80, 27]					
5	[80, 28]					
5	[80, 29]					
5	[80, 30]					
5	[80, 31]					
5	[80, 32]					
5	[80, 33]					
5	[80, 34]					
5	[80, 35]					
5	[80, 36]					
5	[80, 37]					
5	[80, 38]					
5	[80, 39]					
5	[80, 40]					
5	[80, 41]					
5	[80, 42]					
5	[80, 43]					
5	[80, 44]					
5	[80, 45]					
5	[80, 46]					
5	[80, 47]					
5	[80, 48]					
5	[80, 49]					
5	[80, 50]					
5	[80, 51]					
5	[80, 52]					
5	[85, 1]					
5	[90, 1]					
5	[90, 2]					

K	G	W	C	P	S	A
5	[90, 3]					
5	[90, 4]					
5	[90, 5]					
5	[90, 6]					
5	[90, 7]					
5	[90, 8]					
5	[90, 9]					
5	[90, 10]					
5	[95, 1]					
5	[100, 1]	4	28	4	4	4
5	[100, 2]	4	100	4	4	4
5	[100, 3]	4	10	4	4	4
5	[100, 4]	4	28	4	4	4
5	[100, 5]	4	100	4	4	4
5	[100, 6]	4	40	4	4	4
5	[100, 7]	4	28	4	4	4
5	[100, 8]	4	100	4	4	4
5	[100, 9]	4	25	4	4	4
5	[100, 10]	4	13	4	4	4
5	[100, 11]	4	10	4	4	4
5	[100, 12]	4	10	4	4	4
5	[100, 13]	4	16	4	4	4
5	[100, 14]	4	40	4	4	4
5	[100, 15]	4	28	4	4	4
5	[100, 16]	4	100	4	4	4
7	[7, 1]	1	7	1	1	1
7	[14, 1]	2	5	2	2	2
7	[14, 2]	2	14	2	2	2
7	[21, 1]	3	5	3	3	3
7	[21, 2]	3	21	3	3	3
7	[28, 1]	3	10	4	3	2
7	[28, 2]	3	28	4	3	2
7	[28, 3]	4	10	4	4	4
7	[28, 4]	4	28	4	4	4
7	[35, 1]					
7	[42, 1]	6	7	6	6	6
7	[42, 2]	6	10	6	6	6
7	[42, 3]	3	21	3	3	3
7	[42, 4]	6	15	6	6	6
7	[42, 5]	3	12	3	3	3
7	[42, 6]	6	42	6	6	6
7	[49, 1]	1	49	1	1	1
7	[49, 2]	1	49	1	1	1
7	[56, 1]					
7	[56, 2]					
7	[56, 3]					
7	[56, 4]					
7	[56, 5]					
7	[56, 6]					
7	[56, 7]					
7	[56, 8]					

K	G	W	C	P	S	A
7	[56, 9]					
7	[56, 10]					
7	[56, 11]					
7	[56, 12]					
7	[56, 13]					
7	[63, 1]					
7	[63, 2]					
7	[63, 3]					
7	[63, 4]					
7	[70, 1]					
7	[70, 2]					
7	[70, 3]					
7	[70, 4]					
7	[77, 1]					
7	[84, 1]					
7	[84, 2]					
7	[84, 3]					
7	[84, 4]					
7	[84, 5]					
7	[84, 6]					
7	[84, 7]					
7	[84, 8]					
7	[84, 9]					
7	[84, 10]					
7	[84, 11]					
7	[84, 12]					
7	[84, 13]					
7	[84, 14]					
7	[84, 15]					
7	[91, 1]					
7	[98, 1]	2	26	2	2	2
7	[98, 2]	2	98	2	2	2
7	[98, 3]	2	35	2	2	2
7	[98, 4]	2	26	2	2	2
7	[98, 5]	2	98	2	2	2
7	[7, 1]	1	7	1	1	1
7	[14, 1]	2	5	2	2	2
7	[14, 2]	2	14	2	2	2
7	[21, 1]	3	5	3	3	3
7	[21, 2]	3	21	3	3	3
7	[28, 1]	3	10	4	3	2
7	[28, 2]	3	28	4	3	2
7	[28, 3]	4	10	4	4	4
7	[28, 4]	4	28	4	4	4
7	[35, 1]					
7	[42, 1]	6	7	6	6	6
7	[42, 2]	6	10	6	6	6
7	[42, 3]	3	21	3	3	3
7	[42, 4]	6	15	6	6	6
7	[42, 5]	3	12	3	3	3
7	[42, 6]	6	42	6	6	6

K	G	W	C	P	S	A
7	[49, 1]	1	49	1	1	1
7	[49, 2]	1	49	1	1	1
7	[56, 1]					
7	[56, 2]					
7	[56, 3]					
7	[56, 4]					
7	[56, 5]					
7	[56, 6]					
7	[56, 7]					
7	[56, 8]					
7	[56, 9]					
7	[56, 10]					
7	[56, 11]					
7	[56, 12]					
7	[56, 13]					
7	[63, 1]					
7	[63, 2]					
7	[63, 3]					
7	[63, 4]					
7	[70, 1]					
7	[70, 2]					
7	[70, 3]					
7	[70, 4]					
7	[77, 1]					
7	[84, 1]					
7	[84, 2]					
7	[84, 3]					
7	[84, 4]					
7	[84, 5]					
7	[84, 6]					
7	[84, 7]					
7	[84, 8]					
7	[84, 9]					
7	[84, 10]					
7	[84, 11]					
7	[84, 12]					
7	[84, 13]					
7	[84, 14]					
7	[84, 15]					
7	[91, 1]					
7	[98, 1]	2	26	2	2	2
7	[98, 2]	2	98	2	2	2
7	[98, 3]	2	35	2	2	2
7	[98, 4]	2	26	2	2	2
7	[98, 5]	2	98	2	2	2
11	[11, 1]	1	11	1	1	1
11	[22, 1]	2	7	2	2	2
11	[22, 2]	2	22	2	2	2
11	[33, 1]	2	33	3	2	1
11	[44, 1]	3	14	4	3	2
11	[44, 2]	3	44	4	3	2

K	G	W	C	P	S	A
11	[44, 3]	4	14	4	4	4
11	[44, 4]	4	44	4	4	4
11	[55, 1]	5	7	5	5	5
11	[11, 1]	1	11	1	1	1
11	[22, 1]	2	7	2	2	2
11	[22, 2]	2	22	2	2	2
11	[33, 1]	2	33	3	2	1
11	[44, 1]	3	14	4	3	2
11	[44, 2]	3	44	4	3	2
11	[44, 3]	4	14	4	4	4
11	[44, 4]	4	44	4	4	4
11	[55, 1]	5	7	5	5	5
13	[13, 1]	1	13	1	1	1
13	[26, 1]	2	8	2	2	2
13	[26, 2]	2	26	2	2	2
13	[39, 1]	3	7	3	3	3
13	[39, 2]	3	39	3	3	3
13	[52, 1]	4	16	4	4	4
13	[52, 2]	4	52	4	4	4
13	[52, 3]	4	7	4	4	4
13	[52, 4]	4	16	4	4	4
13	[52, 5]	4	52	4	4	4
13	[65, 1]					
13	[78, 1]					
13	[78, 2]					
13	[78, 3]					
13	[78, 4]					
13	[78, 5]					
13	[78, 6]					
13	[91, 1]					
17	[17, 1]	1	17	1	1	1
17	[34, 1]	2	10	2	2	2
17	[34, 2]	2	34	2	2	2
17	[51, 1]	2	51	3	2	1
17	[68, 1]	4	20	4	4	4
17	[68, 2]	4	68	4	4	4
17	[68, 3]	4	8	4	4	4
17	[68, 4]	4	20	4	4	4
19	[19, 1]	1	19	1	1	1
19	[38, 1]	2	11	2	2	2
19	[38, 2]	2	38	2	2	2
19	[57, 1]	3	9	3	3	3
19	[57, 2]	3	57	3	3	3
23	[23, 1]	1	23	1	1	1
23	[46, 1]	2	13	2	2	2
23	[46, 2]	2	46	2	2	2
23	[69, 1]	2	69	3	2	1
23	[92, 1]					
23	[92, 2]					
23	[92, 3]					
23	[92, 4]					

K	G	W	C	P	S	A
29	[29, 1]	1	29	1	1	1
29	[58, 1]	2	16	2	2	2
29	[58, 2]	2	58	2	2	2
29	[87, 1]	2	87	3	2	1
31	[31, 1]	1	31	1	1	1
31	[62, 1]	2	17	2	2	2
31	[62, 2]	2	62	2	2	2
31	[93, 1]	3	13	3	3	3
31	[93, 2]	3	93	3	3	3
37	[37, 1]	1	37	1	1	1
37	[74, 1]	2	20	2	2	2
37	[74, 2]	2	74	2	2	2
41	[41, 1]	1	41	1	1	1
41	[82, 1]	2	22	2	2	2
41	[82, 2]	2	82	2	2	2
43	[43, 1]	1	43	1	1	1
43	[86, 1]	2	23	2	2	2
43	[86, 2]	2	86	2	2	2
47	[47, 1]	1	47	1	1	1
47	[94, 1]	2	25	2	2	2
47	[94, 2]	2	94	2	2	2
53	[53, 1]	1	53	1	1	1
59	[59, 1]	1	59	1	1	1
61	[61, 1]	1	61	1	1	1
67	[67, 1]	1	67	1	1	1
71	[71, 1]	1	71	1	1	1

El caso $\mathbb{F}_2\mathbb{Z}_6$

Otras conjeturas

Nos pareció que sería bueno definir un peso absolutamente irreducible como: 1. Definición. Un **peso absolutamente irreducible** 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 absolutamente irreducible y proyectivo.

Y exploramos la conjetura obvia: "El número de pesos absolutamente irreducibles (hasta conjugación) de KG es igual al número de KG-módulos absolutamente irreducibles (hasta isomorfismo)". Pero $\mathbb{F}_2\mathbb{Z}_6$ es el primero en fallar, y de hecho, no es muy difícil encontrar otros contraejemplos.

Índice alfabético

Bibliografía

[Alperin, 1986] J. L. Alperin. *Local Representation Theory*. Cambridge studies in advanced mathematics. Cambridge University Press, Cambridge; New York (1986).

[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).

[Cabanes, 1988] M. Cabanes. Brauer morphism between modular Hecke algebras. *Journal of Algebra*, 115:1–31 (1988).

[Curtis and Reiner, 1990] Charles W. Curtis and Irving Reiner. *Methods of representation theory–with Applications to Finite Groups and Orders*, volume I of *Wiley classics library of pure and applied mathematics*. Wiley, New York (1990).

[Holt and Rees, 1994] Derek F. Holt and Sarah Rees. Testing modules for irreducibility. *J. Austr. Math. Soc. Ser. A*, 57:1:1–16 (1994).

[Ivanyos and Lux, 2000] Gábor Ivanyos and Klaus Lux. Treating the exceptional cases of the meataxe. *Experimental Mathematics*, 9:373–381 (2000).

[Knörr and Robinson, 1989] R. Knörr and G. R. Robinson. Some remarks on a conjecture of Alperin. *Journal of the London Mathematical Society*, 39:48–60 (1989).

[Okuyama] Okuyama. Unpublished.

[Parker, 1984] R. A. Parker. The computer calculation of modular characters (the Meat-Axe). *Computational Group Theory*, pages 267–274 (1984).

[Schönert and more, 2002] Martin Schönert and more. *GAP – Groups, Algorithms and Programming*. Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hocschule, Aachen, Germany, fourth edition (2002).

[The GAP Group, 2002] The GAP Group. GAP – Groups, Algorithms and Programming. Version 4.3; 2002 (http://www.gap-system.org) (2002).

[Valero-Elizondo, 1998] Luis Valero-Elizondo. *On some invariants associated to simple group representations.* PhD thesis, University of Minnesota (1998).

[Valero-Elizondo, 2002] Luis Valero-Elizondo. Triangular partitions with augmented first rows and weights for the symmetric groups in characteristic two. *Boletín de la Sociedad Matemática Mexicana*, 8:117–126 (2002).