UNIVERSITAT DE VALÈNCIA
FACULTAT DE MATEMÀTIQUES
DEPARTAMENT D'ÀLGEBRA

p-Grupos Finitos

Autor: Ramón Esteban Romero
Director: Dr. D. Antonio Vera López

Memoria presentada para
optar al grado de Doctor en
Ciencias Matemáticas

Valencia, 1997

Índice general

Índice de cuadros

Introducción

La presente memoria se enmarca dentro de la Teoría de los Grupos Finitos. Uno de los objetivos fundamentales de esta teoría es la clasificación, salvo isomorfismo, de estos grupos. Una herramienta fundamental para tratar este problema es la Teoría de Sylow, que establece que todo grupo finito posee p-subgrupos de Sylow para cualquier primo p divisor de su orden. De este modo, se tiene que una aproximación importante a la clasificación de los grupos finitos sería la clasificación de los p-grupos finitos. Sin embargo, este problema presenta una gran dificultad, que es que el número de grupos de orden p^m y la complejidad de sus estructuras aumenta rápidamente según crece el exponente m.

Por este motivo, las investigaciones en el ámbito de la clasificación de los p-grupos finitos se han ido centrando sobre todo en la clasificación de determinadas familias de p-grupos. Por ejemplo, existen teoremas de estructura para los grupos con un subgrupo cíclico de orden p y para los grupos extraespeciales.

Una de las líneas de clasificación de p-grupos, es la clasificación de los mismos por el número de clases de conjugación. El estudio del vector de conjugación de un p-grupo finito nos ha inspirado para la realización de la Parte I, titulada *Algunas cuestiones sobre el número de clases de conjugación de longitud máxima de un p-grupo finito*. En esta parte, se muestra una serie de resultados fruto de mi trabajo conjunto con mi Director, Dr. Antonio Vera López, en el que prestamos una especial atención al número $s = r_G(gN)$ para un subgrupo maximal N del p-grupo finito G de orden p^m y $g \in G \setminus N$, que acotamos entre p^{m-b-1} y $r(G)/p$, encontrando ejemplos en que estas cotas se alcanzan para subgrupos maximales adecuados, y prestamos especial atención al caso en que $s = 1$ para cualquier subgrupo maximal, esto es, el número de clases de conjugación de longitud máxima es lo más grande posible, de la forma $p^{m-b} - 1$. En este caso, localizamos un subgrupo característico de orden p^b e

intentamos encontrar condiciones sobre su estructura. Estudiamos los casos en que $b(G) \in \{1, 2, 3\}$, dando el vector de conjugación de estos grupos.

En la Parte II estudiamos nuevos resultados sobre p-grupos de clase maximal, que son los p-grupos de orden p^m en los que la clase de nilpotencia toma el mayor valor posible, $m - 1$. Los resultados de esta parte han sido obtenidos por el Dr. Antonio Vera López, el Dr. Jesus M. Arregi, M. Asun García Sánchez, el Dr. F. J. Vera López en colaboración conmigo, bajo la dirección del Prof. Antonio Vera López. Esta familia es la que, a priori, presenta una mayor complejidad, pues la serie central tiene longitud máxima, con lo que las relaciones definitorias son más complejas y difíciles de obtener. Las primeras aportaciones a la resolución de este problema corresponden a W. Burnside en [3], quien determina los p-grupos finitos de orden p^4 y clase de nilpotencia 3. Sin embargo, las bases de la teoría de los p-grupos finitos de clase maximal fueron establecidas por A. Wiman en 1946 (véase [37]) y por N. Blackburn en 1958 (véase [2]), quienes introdujeron gran parte de las técnicas de trabajo utilizadas en el cálculo con p-grupos de clase maximal, como la noción de G-sistema generador $(s, s_1, \ldots, s_{m-1})$ (véase 2.1.30) y la serie $Y_i(G)$ (véase 2.1.2). Fue el propio Blackburn el que introdujo el concepto de que es objeto buena parte de esta memoria, el de *grado de conmutatividad* $c(G)$ (véase 2.1.5). Este invariante se define mediante

$$c(G) = \text{máx}\{k \mid k \leq m-2, \quad [Y_i, Y_j] \leq Y_{i+j+k}, \quad i, j \geq 1\}.$$

Claramente, $c(G)$ nos da una idea de en qué medida $Y_1(G)$ (y, en general, los $Y_i(G)$) se aproxima a ser abeliano. Utilizando la noción de G-sistema generador, es posible construir un álgebra de Lie asociada a cada p-grupo de clase maximal, según se ve en la Sección 2.3. En el Capítulo 2 se estudian con más detalle estos conceptos preliminares.

Nuestro objetivo en esta parte es encontrar cotas para el grado de conmutatividad de un p-grupo de clase maximal. Autores como N. Blackburn, R. Shepherd, R. J. Miech, C. R. Leedham-Green o S. McKay han profundizado en las ideas de Blackburn y han centrado su estudio en la obtención de cotas para el grado de conmutatividad del p-grupo G en función del primo p y en dar ejemplos de p-grupos de clase maximal con determinadas propiedades. A lo largo del Capítulo 2 ofrecemos una visión histórica de algunas de las cotas obtenidas para el grado de conmutatividad. La idea central a la hora de obtener estas cotas es tratar de fijar la mayor cantidad posible de invariantes, para poder obtener de una forma más rápida las relaciones definitorias de G fijando previamente dichos invariantes.

El uso de los actuales ordenadores electrónicos permite realizar cálculos repetitivos a gran velocidad y su ayuda nos ha sido de inestimable ayuda a lo largo de toda la memoria. Por ello, dedicamos una parte importante de la misma a la descripción de alguno de los algoritmos que hemos utilizado para el cálculo. Estos algoritmos han sido programados en diversos lenguajes, en virtud de las necesidades de cada momento, como el lenguaje C, GAP [21] o **Maple** [5]. En el Capítulo 3 seguimos las ideas de Shepherd [22] para obtener cotas de carácter general del tipo $2c \geq m - \text{máx}(m-p-2, p-2l-p+c_0-1)$, y precisamos la cota de acuerdo con la tabla 3.3 cuando $l \geq c_0 - 2$. El trabajo de Shepherd permite obtener dicha cota para $0 \leq c_0 \leq 4$, que nosotros extendemos a $0 \leq c_0 \leq 10$ mediante la aplicación de un algoritmo, que considera todas las posibles configuraciones de valores nulos o no nulos que pueden aparecer y aplica el Lema 2.6.1 para obtener dicha cota en la mayoría de las configuraciones; en las configuraciones en las que no sea posible esto utilizamos otro algoritmo basado en la identidad de Jacobi con objeto de obtener la cota y ver que se ajusta a la fórmula deseada. Este algoritmo nos ha permitido obtener los resultados del capítulo 3.

El desarrollo de este último algoritmo nos ha conducido a la elaboración de tablas de doble entrada, con filas correspondientes a los distintos valores de c_0 y columnas correspondientes a los distintos valores de l, en las que, fijado p, obtenemos una cota para $2c$ en función de p, l y c_0. Estas cotas nos permiten dividir la tabla en regiones cuyas casillas satisfacen ciertas relaciones entre p, l y c_0 y para las cuales la función $g(c_0, l, p)$ tal que $2c \geq m - g(c_0, l, p)$ tiene una forma sencilla. Conjeturamos cuáles son los valores de estas funciones y, en algunos casos, damos demostraciones de estas cotas. Para la mayoría de los valores, obtenemos que las cotas son las mejores, ya que construimos álgebras de Lie de clase maximal con $2c = m - g(c_0, l, p)$, como podemos ver en 4.7 para el caso $p = 17$.

En el Apéndice ofrecemos los listados de los programas en C que se han utilizado en la redacción de la presente memoria.

Por último, debo dedicar unas palabras de agradecimiento al Director de esta memoria, Dr. Antonio Vera López, ya que sin sus orientaciones y su constante apoyo no hubiera sido posible este trabajo; a Jesus M. Arregi, M. Asun García y F. J. Vera por su magnífica colaboración en esta aventura investigadora; a los Dres. Daniel Tarazona y José R. Martínez, por su constante ánimo; a Pep Mulet, por sus orientaciones en los temas relacionados con la Informática y, en general, al Departament d'Àlgebra y el Servei d'Informàtica de la Universitat

de València y al Departamento de Matemáticas de la Universidad del País Vasco-Euskal Herriko Unibertsitatea por todas las facilidades que me han prestado. Por último, mi más sincero agradecimiento a mis familiares y mis amigos, por su apoyo y comprensión incluso en los momentos más difíciles.

Esta memoria ha sido realizada al amparo de una Beca Predoctoral de Formación del Personal Investigador de la Generalitat Valenciana, organismo al que deseo mostrar mi agradecimiento más sincero.

Parte I

El vector de conjugación de un p-grupo finito

Capítulo 1

Algunas cuestiones sobre el número de clases de conjugación de longitud máxima de un p-grupo finito

1.1. Definiciones y resultados previos

En todo este capítulo, supondremos que G es un p-grupo finito de orden p^m. Denotaremos por $r_G(S)$ el número de clases de conjugación de G que cortan al subconjunto S, en particular, denotaremos $r(G) = r_G(G)$. Si S es un subconjunto de G, el *vector de conjugación* de S relativo a G se define como

$$\Delta_S^G = (|C_G(g_1)|, |C_G(g_2)|, \ldots, |C_G(g_l)|),$$

donde g_1, $g_2, \ldots$, g_l son las clases de conjugación de G que cortan a S ordenadas de tal manera que $|\mathrm{Cl}_G(g_i)| \leq |\mathrm{Cl}_G(g_{i+1})|$ para todo $i \in \{1, \ldots, l-1\}$. Denotamos, además, $\Delta_G = \Delta_G^G$, que abreviaremos a Δ cuando no haya posibilidad de confusión.

Sea N un subgrupo maximal de G, y $G/N = \langle gN \rangle$. Se tiene entonces el siguiente resultado (ver [24, Theorem 4]):

Proposición 1.1.1. $\Delta_{gN}^G = \Delta_{g^jN}^G = (r_1, \ldots, r_s)$ *para cada* $j = 1, \ldots, p-1$,

y la $r(G)$-tupla

$$(p|C_N(n_1)|,\dots,p|C_N(n_s)|,\overbrace{|C_N(m_1)|,\dots,|C_N(m_t)|}^{t=(r(N)-s)/p},\overbrace{r_1,\dots,r_1}^{p-1},\dots\overbrace{r_s,\dots r_s}^{p-1}),$$

donde $\mathrm{Cl}_N(n_1),\dots,\mathrm{Cl}_N(n_s)$ *son las clases de conjugación de* N *fijadas por* g *y* $\mathrm{Cl}_G(m_1),\dots,\mathrm{Cl}_G(m_t)$ *son el resto de las* G*-clases de conjugación de* N*, es la tupla vector de conjugación excepto el orden de las componentes.*

El número $s = r_G(gN)$ coincide, además, con el número de clases de conjugación de N fijadas por el automorfismo $\psi_g : N \longrightarrow N$ definido por $\psi_g(n) = n^g$ para cada $n \in N$, y se verifica la relación

$$r(G) = ps + \frac{r(N) - s}{p},$$

según el texto de Burnside [3, Note E].

Dado un p-grupo G, denotamos por $b = b(G)$ al número tal que $p^{b(G)}$ es la mayor longitud de una clase de conjugación de G.

Dado un subconjunto normal T de G, denotaremos con $\widehat{r}_G(T)$ el número de clases de conjugación de G contenidas en T de longitud $p^{b(G)}$. Denotaremos por $\widehat{r}(G) = \widehat{r}_G(G)$, el número de clases de conjugación de G de longitud $p^{b(G)}$.

Una observación elemental nos lleva al siguiente resultado:

Proposición 1.1.2. *Si* N *es un subgrupo maximal de* G *y* $g \in G\backslash N$*, entonces se verifica que*

$$\widehat{r}_G(G) = (p-1)\widehat{r}_G(gN) + \widehat{r}_G(N).$$

Demostración. Basta observar que, por la proposición 1.1.1 se tiene que el número de G-clases de conjugación de longitud máxima que aparecen en todos los $r_G(gN)$ son iguales, con lo cual, si expresamos G como unión disjunta de N y de las coclases g^jN con $j \in \{1,\dots,p-1\}$ se tiene el resultado. □

Recordemos también que, según [24, Theorem 2], se verifica que

$$r_G(gN) \equiv 1 \pmod{p-1}$$

para cada $g \in G$ y para cada N subgrupo normal de G.

M. R. Vaughan-Lee probó en [23] el siguiente resultado, que relaciona $b(G)$ con el orden de G', el subgrupo derivado de G.

Teorema 1.1.3 (Vaughan-Lee, [23]). *Si G es un p-grupo con $b = b(G)$, entonces*

$$|G'| \leq p^{\frac{b(b+1)}{2}}.$$

Dicho resultado puede ser combinado con la siguiente relación elemental entre $b(G)$ y $|G'|$:

Lema 1.1.4. *Si G es un grupo y $x \in G$ entonces*

$$|\mathrm{Cl}_G(x)| \leq |G'|.$$

En particular, si G es un p-grupo y $b = b(G)$, concluimos que

$$p^b \leq |G'|.$$

Demostración. Para ver la primera afirmación, construiremos una aplicación inyectiva τ_x entre $\mathrm{Cl}_G(x)$ y $|G'|$. Dicha aplicación vendrá dada por $\tau_x(x^g) = x^{-1}x^g$, τ_x es inyectiva, pues si $x^{-1}x^g = x^{-1}x^h$, $x^g = x^h$. Por tanto, $|\mathrm{Cl}_G(x)| \leq |G'|$.

La segunda afirmación se deduce inmediatamente de la primera, notando que p^b es el cardinal de la mayor clase de conjugación de G. □

Denotemos con $\mathcal{W}_p$ la clase de p-grupos finitos tales que los elementos no centrales tienen el mismo número de conjugados. **Libero Verardi** ha probado los siguientes resultados:

Teorema 1.1.5 (Verardi, [36, 2.3]). *Si $G \in \mathcal{W}_p$, entonces* $\exp G/Z(G) = p$.

Corolario 1.1.6 (Verardi, [36, 2.4]). *Si $G \in \mathcal{W}_2$, entonces G es nilpotente de clase* 2.

Denotamos por $Y_i = Y_i(G)$ los términos de la serie central descendente, esto es, $Y_2 = G'$, $Y_k = [Y_{k-1}, G]$.

Corolario 1.1.7 (Verardi, [36, 2.5]). *Si $p > 2$ y $G \in \mathcal{W}_p$ tiene clase de nilpotencia $c > 2$, entonces $Y_{c-1}(G)$ es abeliano elemental.*

Bert Beisiegel define los *grupos semiextraespeciales (s.e.s.)* como aquellos p-grupos G tales que para cada subgrupo H maximal en $Z(G)$ se tiene que G/H es extraespecial. Recordemos que un p-grupo no abeliano es especial cuando su centro, su derivado y su subgrupo de Frattini coinciden, y es extraespecial cuando es especial y su centro tiene orden p.

Prueba que si $|G| = p^{m+s}$ y $|G'| = p^s$, entonces $m = 2n$, con $n \geq s$, y define los *grupos ultraespeciales (u.s.)* como aquellos p-grupos G para los cuales $s = n$.

Existen ejemplos de grupos u.s. de exponente p y p^2: para los primeros valen los p-subgrupos de Sylow de $SL(3, p^n)$.

Diremos que dos grupos G y H son *isoclínicos* si existen dos isomorfismos

$$\xi\colon G/Z(G) \longrightarrow H/Z(G), \quad \eta\colon G' \longrightarrow H'$$

tales que para todo $x, y \in G$, si se verifican simultáneamente las dos siguientes condiciones,

$$x_1 Z(H) = \big(xZ(G)\big)^{\xi} \quad y_1 Z(H) = \big(yZ(G)\big)^{\xi},$$

entonces $[x, y]^{\eta} = [x_1, x_2]$ (véase, por ejemplo, [13]).

La relación de isoclinismo es una relación de equivalencia en grupos.

Son conocidos los siguientes resultados sobre grupos s.e.s.:

Teorema 1.1.8. *Sea $p > 2$, y $G \in \mathcal{W}_p$. Entonces $b = k$ si, y sólo si, G es isoclínico a un grupo semiextraespecial de exponente p.*

Definimos unos grupos G_n de la siguiente manera. Sea $F_1 = GF(p)$, si definimos para cada $\xi = (x_1, \ldots, x_5)$ y $\eta = (y_1, \ldots, y_t)$ de F_1^5

$$\begin{aligned} \xi\eta \;=\; \big(&x_1 + y_1, x_2 + y_2, x_3 + y_3 + x_2 y_1, \\ &x_4 + y_4 + x_3 y_1 + \frac{1}{2}(x_2 y_1^2 - x_2 y_1), \\ &x_5 + y_5 + x - 3y_2 + x_2 y_1 y_2 + \frac{1}{2}(x_2^2 y_1 - x_2 y_1)\big), \end{aligned}$$

entonces obtenemos un grupo G_1. Entonces sea $F_n = GF(p^n)$: con la misma definición de una operación en F_n^5, obtenemos un p-grupo G_n de orden p^{5n}

en el cual

$$[\xi,\eta] = \big(0,0,x_2,y_1-x_1,y_2,$$
$$x_3y_1 - x_1y_3 + \frac{1}{2}(x_1y_2 - x_2y_1 + x_2y_1^2 - y_2x_1^2),$$
$$x_3y_2 - x_2y_3 + x_2y_1y_2 - x_1x_2y_2$$
$$+ \frac{1}{2}(x_1y_2 - x_2y_1 + x_2^2y_1 - y_2^2x_1)\big).$$

Entonces podemos verificar que

$$(G_n)' = \{(0,0,x_3,x_4,x_5) \mid x_i \in F_n, i = 3,4,5\},$$

y

$$Z(G_n) = \{(0,0,0,x_4,x_5) \mid x_i \in F_n, i = 4,5\}.$$

Más aún, $[G_n : C(\xi)] = p^{2n}$ para cada $\xi \in G_n \setminus Z(G_n)$, con lo cual $G_n \in \mathcal{W}_p$ y, como $Y_3(G_n) = Z(G_n)$, entonces $c = 3$.

Nota 1.1.9. Para el grupo $P = G_1$, tenemos que $m = s + b + 1$ (para m, s, b definidos como en la introducción). Notemos que $\exp P = p$ si $p > 3$ y $\exp P = 9$ si $p = 3$.

Teorema 1.1.10. *Si $p > 2$ y $G \in \mathcal{W}_p$ es tal que $c > 2$ y $s + b + 1 = m$, entonces $c = 3$, $b = 2$ y G es isoclínico al grupo G_1 definido anteriormente.*

1.2. Resultados sobre s

Lema 1.2.1. *Si G es un p-grupo no abeliano, y*

$$G = L_0 > L_1 > \cdots > L_t > L_{t+1} > \cdots > L_n = 1$$

es una serie principal de G, entonces, para cualquier $t \in \{1,\ldots,n-1\}$, se tiene que

$$\widehat{r}(G) = \sum_{i=1}^{t}(p-1)\widehat{r}_G(g_iL_i) + \widehat{r}_G(L_t),$$

donde, para cada $i \in \{1,\ldots t-1\}$, $g_i \in L_{i+1} \setminus L_i$. En particular, tenemos que $\widehat{r}(G) \equiv 0 \pmod{p-1}$.

Demostración. Dado L un subgrupo normal de G y h un elemento de G, denotamos por $T_h = \bigcup_{x\in G} h^x L$ (ver [24]). Evidentemente, se tiene que $\mathrm{Cl}_G(\bar{y}_1) = \mathrm{Cl}_G(\bar{y}_2)$ si, y sólo si, $T_{y_1} = T_{y_2}$.

Consideremos L y L' dos subgrupos normales de G tales que $L \leq L'$ y $|L'/L| = p$. Se tiene entonces que $L'/L \leq Z(G/L)$, con lo que si $h \in L' \setminus L$ y $j \in \{1, \ldots, p-1\}$, entonces $T_h = T_{h^j}$ si, y sólo si, $\mathrm{Cl}_{G/L}(hL) = \mathrm{Cl}_{G/L}(h^j L)$, lo que nos lleva a que $hL = h^j L$, al ser hN un elemento central en G/N. Por tanto, aplicando [24, Theorem 3], deducimos que $\Delta^G_{T_{h^j}} = \Delta^G_{T_h}$, en particular, que $\widehat{r}_G(h^j L) = \widehat{r}_G(hL)$.

Para obtener el resultado, bastará con razonar por inducción sobre t. Desde luego, si $t = 0$, el resultado es claro por (1.1.2). Si la fórmula es válida para un cierto valor de t, tenemos que

$$\widehat{r}(G) = \sum_{i=1}^{t} (p-1)\widehat{r}_G(g_i L_i) + \widehat{r}_G(L_t),$$

con lo que, aplicando el párrafo anterior a $L' = L_t$ y $L = L_{t+1}$, se verifica que

$$\widehat{r}_G(L_t) = (p-1)\widehat{r}_G(g_{t+1}L_{t+1}) + \widehat{r}_G(L_{t+1}),$$

con lo que tenemos probado el resultado para $t+1$, ya que

$$\widehat{r}(G) = \sum_{i=1}^{t+1} (p-1)\widehat{r}_G(g_i L_i) + \widehat{r}_G(L_{t+1}).$$

Notemos que $\widehat{r}_G(L_n) = 0$, con lo que tenemos probada la congruencia. □

En lo sucesivo, denotaremos por $b = b(G)$.

Proposición 1.2.2. *El número s verifica la siguiente acotación:*

$$p^{m-b-1} \leq s \leq \frac{r(G)}{p}.$$

Demostración. Expresemos gN como unión disjunta de clases de conjugación de G, esto es,

$$gN = \mathrm{Cl}_G(y_1) \cup \cdots \cup \mathrm{Cl}_G(y_s).$$

Tenemos entonces que $|\mathrm{Cl}_G(y_i)| \leq p^b$, con lo que se verifica que

$$\frac{|G|}{p} \leq s \cdot p^b,$$

o sea,

$$p^{m-1} \leq sp^b,$$

de donde se obtiene la primera desigualdad

$$p^{m-b-1} \leq s.$$

Para ver la segunda desigualdad, recordemos que se tiene el siguiente resultado:

$$s = r_G(gN) \leq r_G(N) \leq r(N).$$

De esta manera, tenemos que

$$r(G) = (p-1)r_G(gN) + r_G(N) \geq (p-1)r_G(gN) + r_G(gN) = ps,$$

de donde se obtiene que

$$s \leq \frac{r(G)}{p}$$

y tenemos la segunda desigualdad probada. □

Notemos que si se tiene que

$$s = p^{m-b-1},$$

entonces todas las G-clases de conjugación de gN deben ser de cardinalidad máxima p^b. Más en general, se tiene la siguiente acotación de s:

Proposición 1.2.3. *Sea $\widehat{r} = \widehat{r}_G(G)$ el número de clases de conjugación de G de cardinalidad máxima p^b. Tenemos entonces que*

$$s \geq p^{m-b} - \widehat{r}.$$

Demostración. Distinguimos dos casos:
Si $(p-1)p^{m-b-1} \leq \widehat{r}$, entonces tenemos que, por la proposición anterior, se verifica que $s \geq p^{m-b-1}$, y

$$p^{m-b} - \widehat{r} \leq p^{m-b-1} \leq s,$$

puesto que

$$p^{m-b} - p^{m-b-1} = (p-1)p^{m-b-1} \leq \widehat{r}.$$

Si $(p-1)p^{m-b-1} > \widehat{r}$, entonces, en la descomposición

$$gN = \mathrm{Cl}_G(y_1) \cup \cdots \cup \mathrm{Cl}_G(y_s)$$

tiene que haber alguna clase que no sea de longitud maximal, ya que, de otro modo, aparecerían como mínimo

$$p^{m-b-1}(p-1) > \widehat{r}$$

clases de conjugación de longitud maximal. Teniendo en cuenta que para todo $j \in \{1, \ldots, p-1\}$ se tiene que

$$\widehat{r}_G(g^j N) = \widehat{r}_G(gN),$$

tenemos que

$$\widehat{r}_G(gN) \leq \frac{\widehat{r}}{p-1}. \tag{1.1}$$

Los elementos de gN que no están en G-clases de conjugación de gN de longitud maximal serán

$$\frac{|G|}{p} - p^b \widehat{r}_G(gN) = p^{m-1} - p^b \widehat{r}_G(gN),$$

y estarán repartidos en G-clases de conjugación de longitud menor o igual que p^{b-1}. Esto nos da

$$p^{m-b} - p\widehat{r}_G(gN)$$

G-clases de conjugación de longitud no maximal. En total, tenemos que hay, al menos,

$$p^{m-b} - p\widehat{r}_G(gN) + \widehat{r}_G(gN) = p^{m-b} - (p-1)\widehat{r}_G(gN)$$

G-clases de conjugación en gN. Teniendo en cuenta la relación (1.1), tenemos que

$$s = p^{m-b} - (p-1)\widehat{r}_G(gN) \geq p^{m-b} - \widehat{r},$$

que es lo que queríamos demostrar. □

Observemos también que si $ps = r(G)$, entonces $s = r_G(gN) = r_G(N) = r(N)$ y el vector de conjugación de G queda totalmente caracterizado por ser (salvo el orden) la concatenación de p veces el vector de conjugación de N (véase [24, Corollary 2]).

Se conocen grupos en los cuales se alcanzan las cotas anteriores para algún subgrupo maximal.

Ejemplos

1. Supongamos que G es el siguiente grupo:

$$\begin{aligned} G=&((C_8\times C_4)\cdot C_4)\cdot C_4\\ =&((\langle a\rangle\times\langle b\rangle)\cdot\langle c\rangle)\cdot\langle d\rangle \end{aligned}$$

con las relaciones

$$\begin{aligned} a^8=b^4=[a,b]=1,\quad c^2=a^2b^2,\quad d^2=a^4b,\\ a^c=a^d=a^7,\quad b^c=b^3,\quad b^d=b,\quad c^d=cab^3. \end{aligned}$$

Se tiene que el vector de conjugación de G es de la forma

$$(128^2, 64^7, 32^4, 16^8, 8^2).$$

Para los subgrupos maximales

$$M_1=\langle a,b,d\rangle,$$

$$M_2=\langle a,b,cd\rangle,$$

se tiene que $r_G(cM_i)=6$, con lo que se alcanza la cota dada por 1.2.3, puesto que $p^{m-b}=8$ y $\widehat{r}=2$.

2. Consideremos el grupo

$$\begin{aligned} G=&\ C_4\cdot(C_8\cdot(C_2\times C_4\times C_2))=(C_4\cdot(C_8\cdot(C_4\times C_2)))\times C_2\\ =&\langle a_1\rangle\cdot(\langle a_2\rangle\cdot(\langle a_3\rangle\times\langle a_4\rangle\times\langle a_5\rangle))=(\langle a_1\rangle\cdot(\langle a_2\rangle\cdot(\langle a_4\rangle\times\langle a_5\rangle)))\times\langle a_3\rangle \end{aligned}$$

sujeto a las relaciones

$$a_3^2=a_4^4=a_5^2=[a_3,a_4]=[a_3,a_5]=[a_4,a_5]=1,$$

$$a_1^2 = a_5, \quad a_2^2 = a_4, \quad a_3^{a_2} = a_3^{a_1} = a_3,$$
$$a_4^{a_2} = a_4, \quad a_4^{a_1} = a_4^3, \quad a_5^{a_2} = a_5 a_4^4,$$
$$a_5^{a_1} = a_5, \quad a_2^{a_1} = a_2 a_4.$$

El vector de conjugación de dicho grupo es de la forma

$$(128^4, 64^{10}, 32^{10}, 16^8),$$

y se tiene que $r(G) = 32$. Veamos que en este grupo se alcanzan las cotas dadas en (1.2.2):

Si consideramos el subgrupo maximal

$$M_1 = \langle a_2, a_3, a_4, a_5 \rangle$$

se tiene que $r_G(a_1 M_1) = 8$, que es exactamente $p^{m-b-1} = 16/2$. Por otro lado, si consideramos el subgrupo maximal

$$M_2 = \langle a_1, a_3, a_4, a_5 \rangle$$

se tiene que $r_G(a_2 M_2) = 16$, que es $p^{m-1} = 32/2$. Notemos que el vector de conjugación de M_2 es

$$(64^2, 32^5, 16^5, 8^4),$$

esto es, el vector de conjugación de G se obtiene duplicando tanto los órdenes de los centralizadores como el número de clases de cada longitud.

3. Consideremos el grupo

$$\begin{aligned} G=&C_9 \ \cdot(C_3 \ [C_9 \ \times C_3 \ \times C_3]) \\ =&\langle a_1 \rangle \cdot (\langle a_2 \rangle [\langle a_3 \rangle \times \langle a_4 \rangle \times \langle a_5 \rangle]) \end{aligned}$$

con las siguientes relaciones:

$$a_3^9 = a_4^3 = a_5^3 = [a_3, a_4] = [a_3, a_5] = [a_4, a_5] = 1,$$
$$a_2^3 = 1, \quad a_1^3 = a_3^3, \quad a_3^{a_1} = a_3 a_5^2, \quad a_3^{a_2} = a_3^4 a_4^2,$$
$$a_4^{a_1} = a_3^3 a_4, \quad a_4^{a_2} = a_4,$$

$$a_5^{a_1} = a_5, \quad a_5^{a_2} = a_5 a_3^3, \quad a_2^{a_1} = a_2 a_4.$$

El vector de conjugación de este grupo es

$$(729^3, 243^8, 81^{30}, 27^{16}),$$

y $r(G) = 57$. Si consideramos el subgrupo maximal dado por

$$M = \langle a_2, a_3, a_4, a_5 \rangle$$

tenemos que $r_G(a_1 M) = 11 = 27 - 16$, con lo que se alcanza la cota de (1.2.3).

4. Si consideramos el grupo

$$\begin{aligned} G &= C_9 \cdot (C_9 \times C_3 \times C_3 \times C_3) \\ &= \langle a_1 \rangle \cdot (\langle a_2 \rangle \times \langle a_3 \rangle \times \langle a_4 \rangle \times \langle a_5 \rangle) \end{aligned}$$

con las relaciones

$$\begin{aligned} a_2^9 &= a_3^3 = a_4^3 = a_5^3 \\ &= [a_2, a_3] = [a_2, a_4] = [a_2, a_5] \\ &= [a_3, a_4] = [a_3, a_5] = [a_4, a_5] = 1, \end{aligned}$$

$$a_1^3 = a_4, \quad a_2^{a_1} = a_2 a_3^3, \quad a_3^{a_2} = a_3,$$

$$a_4^{a_2} = a_4, \quad a_5^{a_2} = a_5,$$

tenemos que su vector de conjugación es

$$(729^{27}, 243^{72}, 81^{54}).$$

El subgrupo

$$M = \langle a_2, a_3, a_4, a_5 \rangle$$

verifica que $r_G(a_1 M) = 27 = 81/3$.

Corolario 1.2.4. *Se verifica que si G es un p-grupo, entonces existe un entero $k \geq 0$ tal que*

$$s = p^{m-b} - \widehat{r}(G) + k(p-1).$$

Demostración. Sabemos, por (1.1), que se tiene la desigualdad

$$s \geq p^{m-b} - \widehat{r}(G).$$

Además, sabemos que $s \equiv 1 \pmod{p-1}$ (ver [24, Th. 2]), y que, por (1.1.2), se tiene que $\widehat{r}(G) \equiv 0 \pmod{p-1}$. De esta manera, tenemos que

$$s - p^{m-b} + \widehat{r}(G) \equiv 1 - p^{m-b} \pmod{p-1},$$

y, notando que $(p-1) \mid (p^{m-b}-1)$, podemos decir que

$$s - p^{m-b} + \widehat{r}(G) \equiv 0 \pmod{p-1},$$

que es la afirmación que queríamos probar. □

Proposición 1.2.5. *$\widehat{r}(G) \leq p^{m-b}$, donde $b = b(G)$. La igualdad se da si, y sólo si, G es abeliano. En otro caso, se tiene que*

$$\widehat{r}(G) = p^{m-b} - 1 - \lambda(p-1)$$

con $\lambda > 0$.

Demostración. Se tiene que

$$\left|\{g \in G \mid |\mathrm{Cl}_G(g)| = p^b\}\right| \leq |G| = p^m,$$

de donde

$$p^b \cdot \widehat{r}(G) \leq p^m,$$

esto es,

$$\widehat{r}(G) \leq p^{m-b}.$$

Además, se tiene la igualdad si, y sólo si, G es unión de clases de conjugación de longitud máxima, esto es, b es necesariamente 0, o sea, el grupo es abeliano. En otro caso, si G no es abeliano, aplicando el lema, se tiene que

$$\widehat{r}(G) \equiv 0 \pmod{p-1},$$

y

$$p^{m-b} \equiv 1 \pmod{p-1},$$

con lo que

$$\widehat{r}(G) = p^{m-b} - 1 - \lambda(p-1),$$

y al ser $\widehat{r}(G) \leq p^{m-b} - 1$, $\lambda \geq 0$. Esto prueba la segunda afirmación. □

Proposición 1.2.6. *Supongamos que $\widehat{r}(G) = p^{m-b} - 1$. Sea N_b un subgrupo normal de G de orden p^b, donde $b = b(G)$. Entonces N_b es un subgrupo característico de G. Además, si M_b es otro subgrupo normal de G de cardinalidad p^b, entonces $M_b = N_b$.*

Demostración. Notemos que $G \setminus N_b$ está formado exactamente por las clases de conjugación de G que tienen cardinal p^b, ya que no puede haber clases de conjugación de cardinal p^b en N_b y $|G \setminus N_b| = p^m - p^b = p^b(p^{m-b} - 1)$.

Supongamos que M_b es otro subgrupo normal de G de orden p^b. De la misma manera, se tiene que $G \setminus M_b$ está formado exactamente por las clases de conjugación de G de cardinalidad p^b, con lo que $N_b = M_b$. Teniendo en cuenta que N_b es el único subgrupo normal de orden p^b, deducimos que es característico. □

1.3. p-grupos cuyo número de clases de conjugación de longitud máxima es máximo

En esta sección nos proponemos estudiar los p-grupos para los cuales $\widehat{r}(G) = p^{m-b} - 1$, con $b = b(G)$, prestando especial atención al subgrupo normal de orden p^b, N_b (véase la Proposición 1.2.6, ya que todas las clases de conjugación de longitud maximal de G se hallan fuera de N_b.

Definimos la familia $\mathcal{F}$ como la formada por todos los p-grupos G tales que $\widehat{r}(G) = p^{m-b} - 1$.

Deseamos estudiar de la estructura de G/N_b. Conjeturamos que $N_b = Z(G)$. Evidentemente, $Z(G) \leq N_b$, pues las G-clases de conjugación en $Z(G)$ son de longitud 1. Tenemos también que $N_b \leq G'$, en virtud del Lema 1.1.4 y del hecho de ser N_b el único subgrupo normal de G de orden p^b, con lo que deducimos que ha de estar contenido en todo subgrupo normal de mayor orden.

Se han observado todos los 2-grupos de orden menor o igual que 256 y los 3-grupos de orden menor o igual que 729 que verifican esta propiedad. Se han obtenido los grupos que aparecen en las tablas 1.1 y 1.2:

Como se aprecia en las tablas, tenemos que para cada subgrupo maximal de G, el número de clases de conjugación de M fijadas por el automorfismo conjugación, o, si lo preferimos, $r_G(xM)$, con $x \in G \setminus M$, verifica la relación

$$s = p^{m-b-1}.$$

Cuadro 1.1: Los 3-grupos de orden menor o igual que 729 con $\widehat{r}(G)$ máximo

$\|G\|$	#	Δ	m	b	$Z(G)$	$Z_2(G)$	$Z_3(G)$	G'	Y_3	Y_4	G''	G/Z	G/Z_2	G/Z_3	s	x
27	3	$(27^3, 9^8)$	3	1	C_3	$*$		C_3	1		1	C_3^2	1		3	3
27	4	$(27^3, 9^8)$	3	1	C_3	$*$		C_3	1		1	C_3^2	1		3	9
243	3	$(243^9, 27^{26})$	5	2	C_3^2	C_3^3	$*$	C_3^3	C_3^2	1	1	27#3	C_3^2	1	9	9
243	4	$(243^9, 27^{26})$	5	2	C_3^2	C_3^3	$*$	C_3^3	C_3^2	1	1	27#3	C_3^2	1	9	9
243	5	$(243^9, 27^{26})$	5	2	C_3^2	C_3^3	$*$	C_3^3	C_3^2	1	1	27#3	C_3^2	1	9	9
243	6	$(243^9, 27^{26})$	5	2	C_3^2	C_3^3	$*$	C_3^3	C_3^2	1	1	27#3	C_3^2	1	9	9
243	7	$(243^9, 27^{26})$	5	2	C_3^2	C_3^3	$*$	C_3^3	C_3^2	1	1	27#3	C_3^2	1	9	9
243	8	$(243^9, 27^{26})$	5	2	C_3^2	C_3^3	$*$	C_3^3	C_3^2	1	1	27#3	C_3^2	1	9	9
243	9	$(243^9, 27^{26})$	5	2	C_3^2	C_3^3	$*$	C_3^3	C_3^2	1	1	27#3	C_3^2	1	9	9
243	65	$(243^3, 81^{80})$	5	1	C_3	$*$		C_3	1		1	C_3^4	1		27	3
243	66	$(243^3, 81^{80})$	5	1	C_3	$*$		C_3	1		1	C_3^4	1		27	9
729	469	$(729^9, 81^{80})$	6	2	C_3^2	$*$		C_3^2	1		1	C_3^4	1		27	3
729	470	$(729^9, 81^{80})$	6	2	C_3^2	$*$		C_3^2	1		1	C_3^4	1		27	9
729	471	$(729^9, 81^{80})$	6	2	C_3^2	$*$		C_3^2	1		1	C_3^4	1		27	9
729	472	$(729^9, 81^{80})$	6	2	C_3^2	$*$		C_3^2	1		1	C_3^4	1		27	9
729	473	$(729^9, 81^{80})$	6	2	C_3^2	$*$		C_3^2	1		1	C_3^4	1		27	9
729	474	$(729^9, 81^{80})$	6	2	C_3^2	$*$		C_3^2	1		1	C_3^4	1		27	9

Cuadro 1.2: Los 2-grupos de orden menor o igual que 256 con $\widehat{r}(G)$ máximo

$\|G\|$	#	Δ	m	b	$Z(G)$	$Z_2(G)$	$Z_3(G)$	G'	Y_3	Y_4	G''	G/Z	G/Z_2	G/Z_3	s	x
8	3	$(8^2, 4^3)$	3	1	C_2	$*$		C_2	1		1	C_2^2	1		2	4
8	4	$(8^2, 4^3)$	3	1	C_2	$*$		C_2	1		1	C_2^2	1		2	4
32	49	$(32^2, 16^{15})$	5	1	C_2	$*$		C_2	1		1	C_2^4	1		4	4
32	50	$(32^2, 16^{15})$	5	1	C_2	$*$		C_2	1		1	C_2^4	1		4	4
64	241	$(64^4, 16^{15})$	6	2	C_2^2	$*$		C_2^2	1		1	C_2^4	1		8	4
64	242	$(64^4, 16^{15})$	6	2	C_2^2	$*$		C_2^2	1		1	C_2^4	1		8	4
64	243	$(64^4, 16^{15})$	6	2	C_2^2	$*$		C_2^2	1		1	C_2^4	1		8	4
64	244	$(64^4, 16^{15})$	6	2	C_2^2	$*$		C_2^2	1		1	C_2^4	1		8	4
64	245	$(64^4, 16^{15})$	6	2	C_2^2	$*$		C_2^2	1		1	C_2^4	1		8	4
128	2326	$(128^2, 64^{63})$	7	1	C_2	$*$		C_2	1		1	C_2^6	1		32	4
128	2327	$(128^2, 64^{63})$	7	1	C_2	$*$		C_2	1		1	C_2^6	1		32	4
256	55960	$(256^4, 64^{63})$	8	2	C_2^2	$*$		C_2^2	1		1	C_2^6	1		64	4
256	55961	$(256^4, 64^{63})$	8	2	C_2^2	$*$		C_2^2	1		1	C_2^6	1		64	4
256	55962	$(256^4, 64^{63})$	8	2	C_2^2	$*$		C_2^2	1		1	C_2^6	1		64	4
256	55963	$(256^4, 64^{63})$	8	2	C_2^2	$*$		C_2^2	1		1	C_2^6	1		64	4
256	55964	$(256^4, 64^{63})$	8	2	C_2^2	$*$		C_2^2	1		1	C_2^6	1		64	4
256	55965	$(256^4, 64^{63})$	8	2	C_2^2	$*$		C_2^2	1		1	C_2^6	1		64	4
256	55966	$(256^4, 64^{63})$	8	2	C_2^2	$*$		C_2^2	1		1	C_2^6	1		64	4
256	55967	$(256^4, 64^{63})$	8	2	C_2^2	$*$		C_2^2	1		1	C_2^6	1		64	4
256	55968	$(256^4, 64^{63})$	8	2	C_2^2	$*$		C_2^2	1		1	C_2^6	1		64	4

Esta relación se verifica porque en $G \setminus N_b$ están todas las clases de conjugación de G de longitud máxima, y sólo ellas, con lo que, al ser N_b el único subgrupo normal de orden p^b de G, estará contenido en todos los maximales. Por consiguiente, todas las clases de conjugación de $G \setminus M$, para cada subgrupo maximal M, serán de cardinalidad máxima p^b.

1.3.1. Los vectores de conjugación

Proposición 1.3.1. *En todos los casos, $Z(G) \leq N_b$.*

Demostración. Los elementos de $G \setminus N_b$ no serán nunca centrales, ya que sus clases de conjugación tienen cardinal máximo p^b. Por consiguiente, teniendo en cuenta que el único subgrupo normal de orden p^b es N_b, podemos concluir que $Z(G)$ deberá estar contenido en él, ya que no puede tener orden mayor sin contenerlo. □

Los subgrupos maximales no parecen darnos ideas para la inducción. Por ejemplo, el grupo 729#472 tiene sus subgrupos maximales con un vector de conjugación de la forma $(243^9, 81^{24}, 27^{18})$, y un vector de conjugación relativo de la forma $(729^9, 81^{26})$.

Otra observación elemental. Si $b = 1$, tenemos que N_b es un subgrupo normal de orden p, con lo que debe coincidir necesariamente con $Z(G)$, que está contenido en él.

Proposición 1.3.2. *Si $b = 2$, entonces $N_b = Z(G)$.*

Demostración. Supongamos que $b = 2$ y que $N_b \neq Z(G)$. Esto significa que hay elementos en N_b que no están en $Z(G)$, con lo que deben tener clases de conjugación de longitud comprendida entre 1 y p^2 estrictamente, o sea, p. De este modo, el vector de conjugación de G es

$$\left((p^m)^p, (p^{m-1})^{p-1}, (p^{m-2})^{p^{m-2}-1}\right).$$

Entonces

$$r(G) = p^{m-2} + 2p - 2 \equiv |G| \pmod{(p^2-1)(p-1)}.$$

Si escribimos $n = 2m + e$, sabemos que

$$p^m \equiv (p^2-1)m + p^e \pmod{(p^2-1)(p-1)},$$

$$p^{m-2} \equiv (p^2-1)(m-1) + p^e \pmod{(p^2-1)(p-1)},$$

y podemos substituir en la congruencia anterior estos valores para obtener

$$(p^2-1)m + p^e \equiv (p^2-1)(m-1) + p^e + 2p - 2 \pmod{(p^2-1)(p-1)},$$

con lo que

$$p^2 - 1 \equiv 2p - 2 \pmod{(p^2-1)(p-1)}$$

y deducimos que

$$p^2 - 2p + 1 = (p-1)^2 \equiv 0 \pmod{(p^2-1)(p-1)},$$

lo cual es un claro absurdo, ya que $(p^2-1)(p-1) = (p-1)^2(p+1)$. Por consiguiente, si $b = 2$, $N_b = Z(G)$. □

Dado un p-grupo G, se denota por a_i el número de clases de conjugación de G de cardinalidad p^i.

Teorema 1.3.3. *Sea G un p-grupo con $b(G) = 3$ y $\widehat{r}(G) = p^{m-3}-1$. Entonces se tiene una de las siguientes posibilidades:*

- $N_b = Z(G)$, $\Delta_G = \left((p^m)^{p^3}, (p^{m-3})^{p^{m-3}-1}\right)$ *y* $r(G) = p^{m-3} + p^3 - 1$.
- $|G| = p^{2n+1}$, $r(G) = p^{m-3} + p^2 + p - 2$ *y se da una de los dos siguientes casos:*
  - $\Delta_G = \left((p^m)^{p^2}, (p^{m-2})^{p-1}, (p^{m-3})^{p^{m-3}-1}\right)$, *o*
  - $\Delta_G = \left((p^m)^{p}, (p^{m-1})^{p^2-1}, (p^{m-3})^{p^{m-3}-1}\right)$.
- $|G| = p^{2n}$, $\Delta_G = \left((p^m)^{p^2}, (p^{m-1})^{p^2-p}, (p^{m-3})^{p^{m-3}-1}\right)$ *y* $r(G) = p^{m-3} + 2p^2 - p - 1$.

Demostración. Supongamos que no se tiene el caso 1 de la tesis.
Se tiene la igualdad

$$p^3 = |Z(G)| + a_1 p + a_2 p^2,$$

y sabemos que los a_i verifican que

$$a_i \equiv 0 \pmod{p-1}, \quad i = 1, 2.$$

Denotemos $p^t = |Z(G)|$.

- Si $a_2 = p-1$, entonces

$$p^3 - (p^3 - p^2) = p^2 = p^t + a_1 p,$$

con lo cual se dan los siguientes casos:

  - $t = 1$, con lo que $a_1 = \frac{p^2-p}{p} = p-1$ y $r_G(N_b) = 3p-2$.
  - $t = 2$, con lo que $a_1 = 0$ y $r_G(N_b) = p^2 + p - 1$.

- Si $a_2 = 0$, se tiene que $p^3 = p^t + a_1 p$, con lo que tenemos las siguientes posibilidades:

  - $t = 1$, de donde $a_1 = p^2 - 1$ y $r_G(N_b) = p^2 + p - 1$.
  - $t = 2$, de donde $a_1 = p^2 - p$ y $r_G(N_b) = 2p^2 - p$.

Obtenemos, pues, que

$$r_G(N_b) \in \{3p-2, p^2+p-1, 2p^2-p\}.$$

Como $r_G(G \setminus N_b) = \widehat{r}(G) = p^{m-3} - 1$, deducimos que

$$r(G) \in \{p^{m-3} + 3p - 3, p^{m-3} + p^2 + p - 2, p^{m-3} + 2p^2 - p - 1\}.$$

Si escribimos $m = 2n + e$, donde $e \in \{0, 1\}$, tenemos la siguiente igualdad:

$$p^{m-3} = p^{2n+e-3} = p^{2(n-2+e)+1-e},$$

con lo que

$$p^{m-3} \equiv (n-2+e)(p^2-1) + p^{1-e} \pmod{(p^2-1)(p-1)},$$

y

$$r(G) \equiv n(p^2-1) + p^e \pmod{(p^2-1)(p-1)}.$$

Deducimos que

$$r(G) - p^{m-3} \equiv (2-e)(p^2-1) - p^{1-e} + p^e \pmod{(p^2-1)(p-1)}.$$

- Si $a_2 = p-1$, $t = 2$ y $a_1 = 0$, o si $a_2 = 0$, $t = 1$, $a_1 = p^2 - 1$, entonces

$$r(G) - p^{m-3} = p^{m-3} + p^2 + p - 2 - p^{m-3} = p^2 + p - 2,$$

y tenemos que

$$(2-e)(p^2-1) - p^{1-e} + p^e \equiv p^2 + p - 2 \pmod{(p^2-1)(p-1)}.$$

- Si $e = 0$, tenemos que

$$2(p^2-1) - p - p^2 - p + 2 + 1 = p^2 - 2p + 1 = (p-1)^2,$$

y

$$(p-1)^2 \equiv 0 \pmod{(p^2-1)(p-1)},$$

contradicción.

- Si $e = 1$, deducimos que

$$(p^2-1) - 1 + p - p^2 + p + 2 \equiv p - 2 \pmod{(p^2-1)(p-1)},$$

y se llega al caso 2 del enunciado.

- Si $a_2 = p-1$, $t = 1$, $a_1 = p-1$, entonces

$$r(G) - p^{m-3} \equiv p^{m-3} + 3p - 3 - p^{m-3} = 3p - 3 \pmod{(p^2-1)(p-1)},$$

con lo cual

$$(2-e)(p^2-1) + p^e - p^{1-e} \equiv 3p - 3 \pmod{(p^2-1)(p-1)}.$$

- Si $e = 0$, tenemos que

$$2(p^2-1) - p + 1 - 3p + 3 = 2p^2 - 4p + 2 = 2(p-1)^2$$

y

$$2(p-1)^2 \equiv 0 \pmod{(p^2-1)(p-1)},$$

imposible.

- Si $e = 1$, resulta que

$$p^2 - 1 + 1 - 1 + p - 3p - 3 + p^2 - 2p + 1 = (p-1)^2$$

y

$$(p-1)^2 \equiv 0 \pmod{(p^2-1)(p-1)},$$

lo cual es otra contradicción.

- Por último, si $a_2 = 0$, $t = 2$, $a_1 = p^2 - p$, tenemos que

$$r(G) - p^{m-3} \equiv 2p^2 - p - 1 \pmod{(p^2-1)(p-1)},$$

de donde

$$(2-e)(p^2-1) + p^e - p^{1-e} \equiv 2p^2 - p - 1.$$

- Si $e = 0$, tenemos que

  $$2(p^2-1)+1-p-2p^2+p+1 \equiv 0 \pmod{(p^2-1)(p-1)}.$$

  En este caso, $|G| = p^{2n}$, $r(G) = p^{m-3}+2p^2-p-1$, $|Z(G)| = p^2$ y

  $$\Delta_G = \left((p^m)^{p^2}, (p^{m-1})^{p^2-p}, (p^{m-3})^{p^{m-3}-1}\right),$$

  con lo que se cumple la parte 2 de la tesis.
- Si $e = 1$, resulta que

  $$p^2-1+p-1-2p^2+p+1 = -p^2+2p-1 = -(p-1)^2$$

  y

  $$-(p-1)^2 \equiv 0 \pmod{(p^2-1)(p-1)},$$

  lo cual es absurdo. □

Conocemos ejemplos del primero y del tercer tipo. De momento, no conocemos ejemplos en los que los vectores de conjugación tomen los otros dos valores. Por ejemplo, el siguiente grupo tiene su vector de conjugación del tercer tipo:

$$G_1 = C_3\Big[C_3\big[C_3[C_3 \times C_3 \times C_3 \times C_3]\big]\Big] = \langle g_1, g_2, g_3, g_4, g_5, g_6, g_7\rangle$$

con las relaciones

$$\begin{array}{lll} & & g_2^{g_1} = g_2g_5 \\ & g_3^{g_2} = g_3g_5^2g_6 & g_3^{g_1} = g_3g_6 \\ g_4^{g_3} = g_4g_7 & g_4^{g_2} = g_4g_5 & g_4^{g_1} = g_4g_5g_6 \\ g_5^{g_3} = g_5 & g_5^{g_2} = g_5 & g_5^{g_1} = g_5g_7 \\ g_6^{g_3} = g_7 & g_6^{g_2} = g_6g_7 & g_6^{g_1} = g_6g_7^2 \\ g_7^{g_3} = g_7 & g_7^{g_2} = g_7 & g_7^{g_1} = g_7 \\ g_3^3 = 1 & g_2^3 = 1 & g_1^3 = 1 \end{array}$$

tiene como series centrales (tanto ascendente como descendente)

$$\langle g_7\rangle \leq \langle g_5, g_6, g_7\rangle \leq G.$$

En este grupo, $b(G) = 3$, y el subgrupo normal de orden 3^3 es abeliano elemental, $\langle g_5, g_6, g_7\rangle$. $G/Z(G) \cong 729\#469$, y $\Delta_G = (2187^3, 729^8, 81^{80})$.
Los grupos *ultraespeciales* definidos en [1, Lemma 3] son un buen ejemplo de grupos de orden p^{3n} y con $b(G) = n$. La construcción es como sigue:

Lema 1.3.4 (Beisiegel, [1, Lemma 3])**.** *Sea p un primo, L el cuerpo de p^n elementos y K el cuerpo primo de L. En el producto cartesiano $L^{(3)} = P$ definimos un producto como sigue:*

$$(a,b,c)(a',b',c') = \big(a+a', b+b', c+c'+f(a,b')\big),$$

donde f es una aplicación K-lineal de $L \times L$ en L. Se tiene:

- *P es un p-grupo de clase 2.*
- *P es ultraespecial cuando para cada elemento $a \in L \setminus \{0\}$,*

$$\{f(a,l) \mid l \in L\} = \{f(l,a) \mid l \in L\} = L.$$

Lema 1.3.5. *Sea P el grupo definido en el Lema 1.3.4 para el caso especial $f(a,b) = ab$. Entonces:*

- *P es un grupo ultraespecial.*
- *P es isomorfo a un p-subgrupo de Sylow de $SL(3,p^n)$.*
- *P tiene un automorfismo φ de orden $p^n - 1$ que actúa sin puntos fijos sobre P/P' y trivialmente sobre P.*

La siguiente presentación define un grupo G ultraespecial de orden 3^9, para el cual $b(G) = 3$ y $\Delta_G = (19683^{27}, 729^{728})$:

$$\begin{array}{rccc} G = & \langle C_3 \times C_3 \times C_3 \rangle & \times_\varphi & \langle C_3 \times C_3 \times C_3 \times C_3 \times C_3 \times C_3 \rangle \\ = & \langle g_1, g_2, g_3 \rangle & \times_\varphi & \langle g_4, g_5, g_6, g_7, g_8, g_9 \rangle \end{array}$$

con las relaciones siguientes:

$$\begin{array}{lll} g_4^{g_1} = g_4 g_8 g_9^2 & g_4^{g_2} = g_4 g_7 g_8^2 g_9^2 & g_4^{g_3} = g_4 g_7 \\ g_5^{g_1} = g_5 g_7 g_8^2 g_9^2 & g_5^{g_2} = g_5 g_7 & g_5^{g_3} = g_5 g_8 \\ g_6^{g_1} = g_6 g_7 & g_6^{g_2} = g_6 g_8 & g_6^{g_3} = g_6 g_9 \\ g_7^{g_1} = g_7 & g_7^{g_2} = g_7 & g_7^{g_3} = g_7 \\ g_8^{g_1} = g_8 & g_8^{g_2} = g_8 & g_8^{g_3} = g_8 \\ g_9^{g_1} = g_9 & g_9^{g_2} = g_9 & g_9^{g_3} = g_9 \end{array}$$

1.3.2. Estudio de $G/Z(G)$

En todos los casos observados hasta ahora, el centro es abeliano elemental. También se observa que $G/Z(G)$ es un grupo de la familia, o es abeliano elemental. Sin embargo, si N es un subgrupo normal minimal de G, no podemos afirmar que $b(G/N) < b(G)$, como se puede ver en el grupo $G = 243\#3$, para el cual $b(G) = 2 = b(G/N)$ para cualquier subgrupo normal minimal N de G, ni $G/N \in \mathcal{F}$ para ningún N normal minimal. Por tanto, no podemos aplicar directamente el resultado sobre $b(G)$ de "Clases de conjugación de cardinalidad máxima en un p-grupo finito" ([35]). Sin embargo, sí que se tienen los siguientes resultados.

Teorema 1.3.6. *Sea G un p-grupo tal que $\widehat{r}(G) = p^{m-1} - 1$ y $b(G) = 1$. Entonces $Z(G) \cong C_p$ y $G/Z(G)$ es abeliano elemental. Recíprocamente, si G es un p-grupo no abeliano tal que $Z(G) \cong C_p$ y $G/Z(G)$ es abeliano, entonces $b(G) = 1$ y $\widehat{r}(G) = p^{m-1} - 1$.*

Demostración. Consideremos $H = \prod_{x \in G} G/N_x$, donde $N_x = C_G(x)$ (notemos que $C_G(x)$ es siempre un subgrupo normal de G, puesto que tiene índice 1 para los elementos centrales, y tiene índice p para los elementos no centrales, cuya clase de conjugación tiene longitud p). Consideremos la aplicación

$$\begin{array}{rccc} \rho\colon & G & \longrightarrow & H \\ & g & \longmapsto & (gN_x)_{x \in G}. \end{array}$$

Tenemos que

$$\begin{aligned} \operatorname{Ker}\rho &= \{g \in G \mid gN_x = N_x \quad \forall x \in G\} \\ &= \{g \in G \mid g \in N_x \quad \forall x \in G\} \\ &= Z(G), \end{aligned}$$

y, por el Teorema de Isomorfía, se tiene que $G/\operatorname{Ker}\rho = G/Z(G)$ es isomorfo a un subgrupo de H.

El grupo H es un producto de grupos de orden 1 ó p, con lo que es abeliano elemental. Deducimos que $G/Z(G)$ es abeliano elemental.

Recíprocamente, supongamos que $Z(G) \cong C_p$ y $G/Z(G)$ es abeliano y no trivial. Dados $g, h \in G$, tenemos que

$$(g^{-1}g^h)Z(G) = Z(G),$$

con lo cual

$$g^{-1}g^h \in Z(G)$$

y existe $z \in Z(G)$ tal que $g^h = gz$. Por consiguiente, $|\mathrm{Cl}_G(g)| \leq p$, de donde $b(G) = 1$ y $\widehat{r}(G) = p^{m-1} - 1$. □

Corolario 1.3.7. *1. Sea G un p-grupo con $b(G) = 1$ y $\widehat{r}(G) = p^{m-1} - 1$. Entonces G es isomorfo a un producto directo con centro amalgamado de grupos no abelianos G_i con $|G_i/Z(G_i)| = p^2$ y $Z(G_i) = Z(G)$.*

2. Sea G un 2-grupo con $b(G) = 1$ y $\widehat{r}(G) = 2^{m-1} - 1$. Entonces G es un producto directo con centro amalgamado de m copias del grupo diédrico de orden 8, o un producto directo con centro amalgamado de $m - 1$ copias del grupo diédrico de orden 8 y una copia del grupo cuaternio de orden 8, siendo $|G| = 2^{2m+1}$.

Demostración. Véase [12, pp. 353–356, Sätze 13.7, 13.8]. □

Proposición 1.3.8. *Supongamos que $b(G) = 2$ y $\widehat{r}(G) = p^{m-2} - 1$, y que $G/Z(G)$ es un grupo abeliano. Si A es un subgrupo normal minimal de G, entonces $\overline{G} = G/A$ es un grupo con $b(\overline{G}) = 1$ y $\widehat{r}(\overline{G}) = p^{m-1} - 1$.*

Demostración. Si A es un subgrupo normal minimal de G, tenemos que

$$r(G) \geq r(G/A) + \frac{p-1}{p} \cdot |Z(G)|$$

(véase [33, Lemma 2.1]). Sabemos que $r(G) = p^{m-2} + p^2 - 1$ si $|G| = p^m$, y que $|Z(G)| = p^2$. Por tanto,

$$p^{m-2} + p^2 - 1 \geq r(G/A) + (p-1)p,$$

o sea,

$$p^{m-2} + p - 1 \geq r(G/A).$$

Por otro lado, como $Z(G/A) \geq Z(G)/A$, $|Z(G/A)| \geq p$ y $G/Z(G)$ es abeliano, resulta que $G' = Z(G)$ y $(G/A)' = G'/A$ tiene orden p. Deducimos así que $b(\overline{G}) = 1$, ya que la longitud de una clase de conjugación está acotada por el orden del derivado. Denotemos

$$p^z = |Z(G/A)|,$$

entonces

$$r(G/A) = p^z + \frac{p^{m-1} - p^z}{p} = p^{m-2} + p^z - p^{z-1} = p^{m-2} + p^{z-1}(p-1),$$

esto es,

$$r(G/A) \geq p^{m-2} + p - 1,$$

de donde $r(G/A) = p^{m-2} + p - 1$ y $z = 1$, o sea, $\overline{G} = G/A$ es un grupo con $|Z(\overline{G})| = p$, $\overline{G}/Z(\overline{G})$ abeliano y $b(\overline{G}) = 1$, que, por el resultado anterior es un grupo de nuestra familia. □

1.3.3. Cuestiones aritméticas

Otra propiedad común que se observa en los ejemplos es la siguiente: $|G/G'|$ es un cuadrado perfecto. En este sentido, tenemos los siguientes resultados.

Proposición 1.3.9. *Supongamos que $b(G) = 1$ y $\widehat{r}(G) = p^{m-1} - 1$. Entonces $|G/G'|$ es un cuadrado perfecto.*

Demostración. Sabemos que si $b(G) = 1$ y $\widehat{r}(G) = p^{m-1} - 1$, entonces $|G'| = |Z(G)| = p$, y que

$$r(G) = p^{m-1} + p - 1.$$

Escribamos $|G| = p^m$, con $m = 2n + e$ y $e \in \{0, 1\}$. Como $|G'| = p$, decir que $|G/G'|$ es un cuadrado perfecto, equivale a decir que $e = 1$. Supongamos, pues, que $e = 0$ y veamos que se llega a un absurdo.

Es conocido que

$$r(G) \equiv |G| \pmod{(p^2-1)(p-1)},$$

y que

$$p^m = p^{2n} \equiv n(p^2-1) + 1 \pmod{(p^2-1)(p-1)},$$
$$p^{m-1} = p^{2(n-1)+1} \equiv (n-1)(p^2-1) + p \pmod{(p^2-1)(p-1)},$$

con lo que

$$n(p^2-1) + 1 \equiv (n-1)(p^2-1) + p + p - 1 \pmod{(p^2-1)(p-1)},$$

de donde

$$1 \equiv -p^2 + 1 + 2p - 1 \pmod{(p^2-1)(p-1)},$$

esto es,

$$0 \equiv -(p-1)^2 \pmod{(p^2-1)(p-1)}.$$

Esto es absurdo, con lo que $e = 1$ y $|G/G'|$ es un cuadrado perfecto. □

Proposición 1.3.10. *Supongamos que $b(G) = 2$ y $\widehat{r}(G) = p^{m-2} - 2$. Supongamos que se da, además, uno de los dos siguientes casos:*

1. *$\overline{G} = G/Z(G)$ es un grupo abeliano.*
2. *$\overline{G} = G/Z(G)$ es un grupo no abeliano con $b(\overline{G}) = 1$, $\widehat{r}(\overline{G}) = p^{m-2} - 1$ (esto es, un grupo no abeliano de $\mathcal{F}$).*

Entonces $|G/G'|$ es un cuadrado perfecto.

Demostración. Si $G/Z(G)$ es abeliano, sabemos, por 1.3.8, que si A es un normal minimal de G, $G/A \in \mathcal{F}$ y $b(G/A) = 1$, y por 1.3.9, $(G/A)\big/(G/A)' \cong G/G'$ tiene orden cuadrado perfecto.

Supongamos ahora que $G/Z(G) \in \mathcal{F}$ es un grupo no abeliano. Sabemos, por un resultado de Vaughan-Lee ([23]) que $|G'| \leq p^{b(G)\left(b(G)+1\right)/2}$, con lo que, en este caso, $|G'| = p^3$ y, por estar la longitud de cada clase de conjugación de $G/Z(G)$ acotada por el orden de su subgrupo derivado, y al ser el orden de $\big(G/Z(G)\big)' = G'/Z(G)$ igual a p, deducimos que $\big(G/Z(G)\big)\big/\big(G/Z(G)\big)' \cong G/G'$ tiene orden cuadrado perfecto, en virtud de 1.3.9. □

Podemos afinar aún más el razonamiento anterior.

Proposición 1.3.11. *Supongamos que $b(G) = 2$, y que $\widehat{r}(G) = p^{n-2} - 1$. Entonces $G/Z(G)$ es abeliano elemental y su orden es un cuadrado perfecto, o $G/Z(G)$ es isoclínico a un grupo extraespecial de exponente p y $|G/Z_2(G)|$ es un cuadrado perfecto.*

Demostración. Si $G/Z(G)$ es abeliano, ya hemos visto que es abeliano elemental. Supongamos que $G/Z(G)$ no es abeliano elemental. Se tiene que el orden del subgrupo derivado de G, G', es menor o igual que p^3. De este modo,

$$|G'/Z(G)| = \left|\big(G/Z(G)\big)'\right| \leq p.$$

Esto nos lleva a que $b\big(G/Z(G)\big) \leq 1$, ya que si x, $y \in H = G/Z(G)$, $x^y = x[x,y]$, con lo cual $|\mathrm{Cl}_H(x)| \leq p$. Por tanto, H es un grupo en el cual las clases de conjugación no centrales tienen la misma longitud. Según [36], H es isoclínico a un grupo extraespecial de exponente p, y ya hemos visto que G tiene exponente p.

Además, $G/Z(G) \cong H/Z(G)$, con lo cual, al ser H isoclínico a un grupo extraespecial, $|H/Z(H)|$ es un cuadrado y, por tanto, $|G/Z_2(G)|$ es un cuadrado perfecto. □

El anterior razonamiento puede ser generalizado como se ve en la Proposición 1.3.13. Notemos primero el siguiente resultado:

Lema 1.3.12. *Sea G un p-grupo de orden p^m. Supongamos que $\widehat{r}(G) = b-1$, donde $b(G) = b$. Entonces, si $G/Z(G)$ es abeliano, entonces $r(G) = p^{m-b} + p^b - 1$. Además, si $r(G) = p^{m-b} + p^b - 1$, podemos concluir que $N_b = Z(G)$.*

Demostración. Según hemos observado en la introducción de esta sección, se tiene que $Z(G) \leq N_b \leq G'$. Por tanto, $G/Z(G)$ es abeliano si, y sólo si, $Z(G) = N_b = G'$. Si $N_b = Z(G)$, se tiene que $\widehat{r}(G) = p^{m-b} - 1$, y $r_G\big(Z(G)\big) = p^b$, de donde $\widehat{r}(G) = p^{m-b} + p^b - 1$.

Por otro lado, si $\widehat{r}(G) = p^{m-b} - 1$ y $r(G) = p^{m-b} + p^b - 1$, tenemos que $r_G(N_b) = p^b$, o sea, todas las clases de conjugación de N_b son centrales. Como $Z(G) \leq N_b$, tenemos que $Z(G) = N_b$. □

Proposición 1.3.13. *Sea G un p-grupo de orden p^m con $b(G) = b \geq 3$, y $\widehat{r}(G) = p^{m-b} - 1$. Supongamos que $G/Z(G)$ es un grupo abeliano. Entonces, si A es un subgrupo normal minimal de G, el grupo $\overline{G} = G/A$ es un grupo con $b(\overline{G}) = b - 1$ y $\widehat{r}(\overline{G}) = p^{m-b} - 1$. Además, en este caso, $|G/G'|$ es un cuadrado perfecto.*

Demostración. Escribamos $p^z = |Z(\overline{G})|$, como $|Z(G)| = p^b$ (ya que $Z(G) \leq N_b \leq G'$ y $G' \leq Z(G)$ por ser $G/Z(G)$ abeliano), tenemos que $z \geq b-1$. Sea a_i el número de clases de conjugación de $\overline{G}$ de cardinalidad p^i. Entonces

$$r(\overline{G}) = p^z + \sum_{i=1}^{b-1} a_i,$$

teniendo en cuenta que $b(\overline{G}) \leq b-1$ porque $|\overline{G}'| = p^{b-1}$. Además, se tiene que

$$|\overline{G}| = p^{m-1} = p^z + \sum_{i=1}^{b-1} a_i p^i,$$

con lo que podemos despejar a_{b-1} como

$$a_{b-1} = \frac{p^{m-1} - p^z - \sum_{i=1}^{b-2} a_i p^i}{p^{b-1}} = p^{m-b} - p^{z-b+1} - \sum_{i=1}^{b-2} a_i p^{i-b+1},$$

y, por consiguiente,

$$r(\overline{G}) = p^z + p^{m-b} - p^{z-b+1} + \sum_{i=1}^{b-2} a_i(1 - p^{i-b+1}),$$

esto es,

$$r(\overline{G}) = p^{m-b} + p^{z-b+1}(p^{b-1} - 1) + \sum_{i=1}^{b-2} a_i(1 - p^{i-b+1}).$$

De este modo, aplicando el Lema 1.3.12,

$$r(\overline{G}) \geq p^{m-b} + p^{b-1} - 1, \tag{1.2}$$

y se tiene la igualdad si, y sólo si, todos los a_i, para $i = 1, \ldots, b-2$ son iguales a cero y $z - b + 1 = 0$, o sea, $z = b - 1$.

Por otro lado, consideremos el resultado de [33, Lemma 2.1]. Entonces

$$r(G) \geq r(G/A) + \frac{p-1}{p}|Z(G)|,$$

con lo que

$$r(G/A) \leq p^{m-b} + p^b - 1 - (p-1)p^{b-1} = p^{m-b} + p^{b-1} - 1. \tag{1.3}$$

y, de 1.2 y 1.3 se obtiene que

$$r(\overline{G}) = p^{m-b} + p^{b-1} - 1,$$

y la desigualdad 1.2 se cumple con igualdad, de donde $Z(\overline{G})$ tiene orden p^{b-1} y todos los a_i, para $1 \leq i \leq b-2$ valen 0.

La segunda afirmación ($|G/G'|$ es un cuadrado perfecto) se obtiene inmediatamente mediante un razonamiento inductivo considerando cocientes por subgrupos normales minimales. □

1.3.4. Estudio del exponente de estos grupos

Algunos de los ejemplos encontrados hasta ahora de grupos con la propiedad $\widehat{r}(G)$ máximo verifican que $|Z(G)| = p^b$. Se observa la siguiente relación entre b y m en todos estos casos:

Proposición 1.3.14. *Supongamos que $G \in \mathcal{F}$ y que $|Z(G)| = p^b$. Entonces $2b \leq m-1$. Además, si $x \in G \setminus Z(G)$, $o\big(xZ(G)\big) \leq p^{m-2b}$.*

Demostración. En las hipótesis del enunciado, si $x \notin Z(G)$, $|C_G(x)| = p^{n-b}$, ya que $N_b = Z(G)$. De esta manera, teniendo en cuenta que

$$\langle x, Z(G)\rangle \leq C_G(x),$$

deducimos que

$$|Z(G)|p = p^{b+1} \leq p^{m-b},$$

de donde

$$b+1 \leq m-b,$$

que es justamente la primera afirmación. La segunda afirmación sale como consecuencia de que

$$o\big(xZ(G)\big) = \big|\langle x, Z(G)\rangle\big|/|Z(G)| \leq |C_G(x)|/|Z(G)| = p^{m-2b}$$

si $x \notin N_b = Z(G)$. □

Otra de las regularidades observadas en la estructura de los grupos obtenidos con la condición $\widehat{r}(G) = p^{n-b} - 1$ y con $G/Z(G)$ abeliano es que tanto $G/Z(G)$ como $Z(G)$ son grupos abelianos elementales. Probamos los siguientes resultados:

Proposición 1.3.15. *Sea G un p-grupo con $b = b(G)$, $\widehat{r}(G) = p^{m-b} - 1$ y $G/Z(G)$ abeliano. Entonces $G/Z(G)$ es abeliano elemental.*

Demostración. Teniendo en cuenta las Proposiciones 1.3.8 y 1.3.13, si consideramos M un normal minimal de G, G/M es un grupo de la familia $\mathcal{F}$ con $b(G/M) = b(G) - 1$. Esto nos permite utilizar inducción sobre $b = b(G)$ de la siguiente manera:

- Si $b = 1$, sabemos, por el Teorema 1.3.6 que $G/Z(G)$ es siempre abeliano elemental.

- Supongamos el resultado válido para grupos H de $\mathcal{F}$ con $b(H) = b - 1$ y $H/Z(H)$ abeliano, y veamos que lo es para un grupo G de $\mathcal{F}$ con $G/Z(G)$ abeliano. Sabemos, por la Proposición 1.3.13, que si M es un normal minimal de G y $G/Z(G)$ es abeliano, entonces $H = G/M$ es un grupo de $\mathcal{F}$ con $b(H) = b(G) - 1$. Por otro lado, $Z(G) \leq N_b \leq G'$, y como $G/Z(G)$ es abeliano, se tiene que $G' = Z(G)$. Como $M \leq G'$, tenemos que $H' = G'/M$, o sea, $H/H' \cong G/G'$. Además, es claro que $Z(G)/M \leq Z(H)$, y $Z(H) \leq N_{b-1}(H) \leq H'$, con lo que de nuevo se tiene la igualdad $Z(H) = H'$ y $H/Z(H)$ es abeliano. Podemos aplicar la hipótesis de inducción, con lo que H/H' es abeliano elemental y, por consiguiente, G/G' lo es. □

Corolario 1.3.16. *Si G es un p-grupo con $b = b(G)$, $\widehat{r}(G) = p^{m-b} - 1$ y $G/Z(G)$ abeliano, entonces $\Phi(G) = G'G^p = G' = N_b = Z(G)$ (o sea, G es especial).*

Demostración. Basta observar que $\Phi(G)$ es el menor subgrupo de G que da cociente abeliano elemental, y que $G/Z(G) = G/G'$ es abeliano elemental. □

En relación a la estructura de $Z(G)$ y al exponente de G cuando $G/Z(G)$ es abeliano, se han obtenido los siguientes resultados:

Lema 1.3.17. *Supongamos que G es un p-grupo con $b = b(G)$, $\widehat{r}(G) = p^{m-b} - 1$ y $G/Z(G)$ abeliano. Dado un conmutador $[a, b] = a^{-1}b^{-1}ab$, con a, $b \in G$, se tiene que $[a, b]^p = 1$.*

Demostración. Veamos, por inducción, que

$$a^{b^r} = a[a, b]^r. \tag{1.4}$$

Para $r = 1$, el resultado se reduce a la conocida igualdad

$$a^b = a[a, b].$$

Supongamos que el resultado es cierto para $r-1$, veámoslo para r.

$$\begin{aligned} a^{b^r} &= \left(a^{b^{r-1}}\right)^b \\ &= \left(a[a,b]^{r-1}\right)^b \\ &= a^b\left([a,b]^b\right)^{r-1} \\ &= a[a,b][a,b]^{r-1} \qquad (1.5) \\ &= a[a,b]^r, \end{aligned}$$

donde en (1.5) hacemos uso del hecho que $[a,b] \in G' = Z(G)$, por el Corolario 1.3.16. Esto prueba (1.4).

Si particularizamos (1.4) al caso $r = p$, obtenemos que

$$a^{b^p} = a[a,b]^p,$$

y como $b^p \in Z(G)$ por ser $G/Z(G)$ abeliano elemental por la Proposición 1.3.15, deducimos que

$$a = a[a,b]^p,$$

o sea,

$$[a,b]^p = 1,$$

que es la afirmación que deseábamos probar. □

Corolario 1.3.18. *Supongamos que G es un p-grupo con $b = b(G)$, $\widehat{r}(G) = p^{m-b} - 1$ y $G/Z(G)$ abeliano. Entonces $Z(G)$ es abeliano elemental.*

Demostración. Sabemos, por el Corolario 1.3.16, que $Z(G) = G'$, y G' viene engendrado por los conmutadores de la forma $[a,b]$, con $a, b \in G$. Ahora bien, $G' = Z(G)$ es abeliano, y sus generadores tienen orden a lo sumo p, con lo que G' tendrá exponente p. El único p-grupo abeliano de exponente p es el grupo abeliano elemental. □

Corolario 1.3.19. *Supongamos que G es un p-grupo con $b = b(G)$, $\widehat{r}(G) = p^{m-b} - 1$ y $G/Z(G)$ abeliano. Entonces* $\exp G \leq p^2$.

Demostración. Sabemos, por la Proposición 1.3.15 que $G/Z(G)$ es abeliano elemental, y, por el Corolario 1.3.18, que $Z(G)$ es abeliano elemental. De este modo, dado $x \in G$, $x^p \in Z(G)$, y $x^{p^2} = (x^p)^p = 1$. Por tanto, $\exp G \leq p^2$. □

En este trabajo ("Clases de conjugación de cardinalidad máxima en un p-grupo finito", [35, pág. 4]), se tiene que si N es un subgrupo normal de G y $g \in G$, $b(gN) < b(g)$ si, y sólo si, se satisface una de las condiciones siguientes:

1. $g \notin C_G(N)$,
2. $C_G(g)N < N_G(gN)$.

Si hacemos $N = Z(G)$, la primera condición no se puede cumplir nunca, pues $C_G(Z(G)) = G$. De este modo, $b\big(gZ(G)\big) < b(g)$ si, y sólo si, $C_G(g)Z(G) < N_G\big(gZ(G)\big)$, o sea,

$$C_G(g) < N_G\big(gZ(G)\big).$$

1.4. Conjeturas y problemas abiertos

1.4.1. Conjetura "class-breadth"

¿Es $b \geq c-1$? La respuesta a esta pregunta es no, según consta en el trabajo de W. Felsch, J. Neubüser y W. Plesken [6]. En este artículo se construye una familia de 2-grupos G_z, con $z \in \mathbb{N}$, tales que

$$c(G_z) \geq b(G_z) + z,$$

que suponen un contraejemplo a la conjetura $b \geq c - 1$. No obstante, aún no se conocen contraejemplos para p-grupos con p impar. En este sentido, se conocen los siguientes resultados:

Teorema 1.4.1 (Leedham-Green, Neumann y Wiegold, [17]). *Las siguientes afirmaciones son válidas:*

1. *Supongamos que G es un p-grupo de clase de nilpotencia c y denotemos $b = b(G)$, entonces*
$$c < \frac{p}{p-1}b + 1.$$
2. *Si G es un p-grupo metaabeliano con $b = b(G)$ y clase de nilpotencia c, entonces*
$$c \leq b + 1.$$

Teorema 1.4.2 (Josep A. Gallian, [8]). *Sea G un p-grupo de orden p^n con $b = b(G)$ y clase de nilpotencia c.*

1. *Si $b \leq c+1$ y $K = \bigcup_{i=1}^{b-2} C_{c-i}$ es tal que $|K| \leq p^n - p^{n-1}$, entonces $c \leq b+1$.*
2. *Si $b \leq p+1$, entonces $c \leq b+1$ (**Theorem 1**).*
3. *Si $c \leq p+3$, entonces $c \leq b+1$.*
4. *Si $p+1 \leq b \leq c+p-1$ y, además, $Y_{c-b+p} \leq Z(Y_2)$, entonces $c \leq b+1$ (**Theorem 2**).*
5. *Si $Y_{p+2} \leq Z(Y_2)$, entonces $c \leq b+1$.*
6. *Si Y_2 puede ser generado por 2 elementos, entonces $c \leq b+1$.*
7. *Si $b > 1$, entonces*
$$c < \frac{p(b+1)-2}{p-1}$$
*(**Theorem 3**).*
8. *Si $b > 1$, entonces $c < 2b$, y $c = 2b$ si $b = 1$.*

Teorema 1.4.3 (M. Cartwright, [4]). *Sea P un p-grupo de clase c tal que $b(G) = b$. Entonces*
$$c \leq \frac{5}{3}b + 1.$$

1.4.2. Conjeturas sobre grupos con $\widehat{r}(G) = p^{m-b} - 1$.

La proposición 1.3.3 establece condiciones restrictivas sobre el vector de conjugación para $b = 3$. La respuesta a la primera pregunta debe buscarse entre grupos que verifiquen las condiciones impuestas por las Proposiciones 1.3.8, 1.3.13 y 1.3.14, si $G/Z(G)$ es abeliano.

¿Podemos asegurar que todo grupo de $\mathcal{F}$ con $b \geq 1$ tiene un cociente por el centro abeliano elemental o en $\mathcal{F}$? Para $b = 2$, nos falta por probar esto en el caso en que $G/Z(G)$ no es abeliano. De todos modos, hemos visto que esto es así salvo isoclinismo.

¿Es siempre $G/Z(G)$ un grupo de exponente p?

Debemos hacer constatar la dificultad en buscar ejemplos con estas condiciones, ya que el grupo factor derivado debe tener un rango bastante grande, lo que hace que el *Algoritmo de Generación de p-Grupos* (véase [19], [20]) no sea una herramienta eficiente para su búsqueda.

Parte II

p-grupos de clase maximal

Capítulo 2

Preliminares

2.1. Conceptos básicos

En todo lo siguiente, p denotará un número primo impar.

Definición 2.1.1. Llamaremos *p-grupo de clase maximal* a un p-grupo G de orden p^m, con $m \geq 4$, y clase de nilpotencia $m-1$.

La teoría de los p-grupos de clase maximal se inició con **Blackburn**, en su trabajo [2]. En este artículo se utilizan algunos resultados probados por **Hall** en [10] y [11].
A continuación definimos algunos subgrupos importantes de los p-grupos de clase maximal.

Definición 2.1.2. Dado un p-grupo de clase maximal G, se define la siguiente serie de subgrupos:

1. Para $i \geq 2$, $Y_i = Y_i(G)$ será el término de la serie central descendente, dada por $Y_2(G) = G' = [G, G]$ y, para $i \geq 2$, $Y_{i+1}(G) = [Y_i(G), G]$.

2. Para $i = 1$, $Y_1 = Y_1(G)$ se define por medio de la relación $Y_1/Y_4 = C_{G/Y_4}(Y_2/Y_4)$.

3. Para $i = 0$, definimos $Y_0(G) = G$.

Veamos a continuación que la serie

$$G = Y_0 > Y_1 > Y_2 > \cdots > Y_{m-1} > Y_m = 1 \tag{2.1}$$

es una cadena de subgrupos característicos de G en la que el índice de cada subgrupo en el anterior es siempre p.

Denotaremos por $Z_i(G)$ los términos de la serie central ascendente, dada por $Z_1(G) = Z(G)$ y $Z_i(G)/Z_{i-1}(G) = Z\big(G/Z_{i-1}(G)\big)$ para $i \geq 2$.

Lema 2.1.3. *Sea G un p-grupo de clase maximal y orden p^m. Entonces son ciertas las siguientes afirmaciones:*

1. *$|G/Y_2(G)| = p^2$ y $|Y_i(G)/Y_{i+1}(G)| = p$ para $2 \leq i \leq m-1$.*
2. *Para todo $i \geq 2$, $Y_i(G)$ es el único subgrupo normal de índice p^i.*
3. *Si $N \trianglelefteq G$ y $|G/N| \geq p^2$, entonces G/N es un p-grupo de clase maximal.*
4. *Se tiene que $Z_i(G) = Y_{m-i}(G)$ para $0 \leq i \leq m-2$.*

Demostración. 1. Esta afirmación se sigue directamente de la igualdad

$$|G| = p^m = \prod_{i=2}^{m-1} |Y_i(G)/Y_{i+1}(G)| \cdot |G/Y_2(G)|$$

y de las desigualdades $|G/Y_2(G)| \geq p^2$ y $|Y_i(G)/Y_{i+1}(G)| \geq p$ para $2 \leq i \leq m-1$.

2. Sea $N \trianglelefteq G$ con $|G/N| = p^i$. Entonces G/N es un p-grupo de clase a lo sumo $i-1$. Por tanto, $1 = Y_i(G/N) = Y_i(G)N/N$, y, de este modo, $Y_i(G) \leq N$. Como tenemos que $|G/Y_i(G)| = p^i$, podemos concluir que $N = Y_i(G)$.

3. Sea $|G/N| = p^i \geq p^2$. Del apartado anterior podemos concluir que $N = Y_i(G)$. Así, pues, G/N tiene clase $i-1$ y, consecuentemente, es un p-grupo de clase maximal (nótese que para $2 \leq j \leq i-1$ se tiene:

$$\begin{aligned} Y_j(G/N)/Y_{j+1}(G/N) &= \big(Y_j(G)N/N\big)\big/\big(Y_{j+1}(G)N/N\big) \\ &= \big(Y_j(G)/N\big)\big/\big(Y_{j+1}(G)/N\big) \\ &= Y_j(G)/Y_{j+1}(G), \end{aligned}$$

de orden p, y también se tiene que

$$\big|(G/N)/Y_2(G/N)\big| = \big|(G/N)/\big(Y_2(G)/N\big)\big| = |G/Y_2(G)| = p^2,$$

como queríamos ver).

4. Como G es de clase $m-1$, se sigue que $Z_{m-1}(G) = G$, siendo $Z_0(G) = 1$, $Z_1(G) = Z(G)$ y $Z_i(G)/Z_{i-1}(G) = Z\big(G/Z_{i-1}(G)\big)$ para $i \geq 2$. Tenemos entonces la cadena

$$1 = Z_0(G) < Z_1(G) < \cdots < Z_{m-1}(G) = G$$

y $|G/Z_{m-2}(G)| \leq p^2$, por tanto, necesariamente $|Z_i(G)| = p^i$ para $i \leq m-2$, y según el segundo apartado, podemos concluir que $Z_i(G) = Y_{m-1}(G)$. □

Lema 2.1.4. *Sea G un p-grupo de clase maximal y orden p^m, con $m \geq 4$. Entonces se verifican las siguientes afirmaciones:*

1. $|G/Y_1| = p$.
2. $Y_1/Y_3 \not\leq Z(G/Y_3)$.

Demostración. 1. Tenemos que existe un monomorfismo entre el grupo cociente $(G/Y_4)/C_{G/Y_4}(Y_2/Y_4)$ y el grupo de automorfismos $\mathrm{Aut}(Y_2/Y_4)$, por tanto, $|G/Y_1|$ debe dividir a $|\mathrm{Aut}(Y_2/Y_4)|$. Este último número coincide con $p(p-1)$ o con $(p^2-1)(p^2-p)$, según sea Y_2/Y_4 isomorfo a C_{p^2} o a $C_p \times C_p$. De este modo, $|G/Y_1| \leq p$.

Si $Y_1 = G$, se tiene que $[G/Y_4, Y_2/Y_4] = 1$, luego $[G, Y_2] = Y_3 \leq Y_4$, lo cual es imposible. Por tanto, $Y_1 < G$ y, necesariamente, $|G/Y_1| = p$, como queríamos ver.

2. Supongamos que $Y_1/Y_3 \leq Z(G/Y_3)$. Tenemos que $(G/Y_3)/(Y_1/Y_3) = C_p$ y la inclusión anterior fuerzan que G/Y_3 sea abeliano, luego $G' = Y_2 \leq Y_3$, una contradicción. □

Definición 2.1.5. El *grado de conmutatividad* de un p-grupo de clase maximal G, $c = c(G)$, viene definido por

$$c(G) = \text{máx}\{k \mid k \leq m-2, \quad [Y_i, Y_j] \leq Y_{i+j+k}, \quad i, j \geq 1\}.$$

Puesto que $[Y_i, Y_j] \leq Y_{i+j}$ para i, $j \geq 1$, se sigue que $c(G) \geq 0$ en cualquier caso.

Definición 2.1.6. Sea G un p-grupo de clase maximal y orden p^m con $m \geq 5$. Decimos que G es un *p-grupo excepcional* si existe i tal que $3 \leq i \leq m-2$ con $Y_1 \neq C_G(Y_i/Y_{i+2})$.

Veremos que el calificativo de excepcional que reciben estos grupos está justificado en función de algunas de las propiedades de estos grupos. También vamos a caracterizar los p-grupos excepcionales como aquellos en los que $c(G) = 0$. Para ello, necesitaremos algunos resultados previos.

Lema 2.1.7. *Sea G un p grupo de clase maximal de orden p^m, con $m \geq 4$. Supongamos que G no es excepcional. Entonces definimos elementos s, s_1 de G tales que $G = \langle Y_1, s\rangle$, $Y_1 = \langle Y_2, s_1\rangle$ y definimos recursivamente elementos s_i, con $i \geq 2$, como sigue: $s_i = [s_{i-1}, s]$. Entonces se tiene que $Y_i = \langle Y_{i+1}, s_i\rangle$ para $2 \leq i \leq m-1$.*

Demostración. Tenemos que $G = \langle Y_1, s\rangle = \langle Y_2, s_1, s\rangle$ y que $Y_2 = G' \leq \Phi(G)$, el subgrupo de Frattini de G, luego $G = \langle s, s_1\rangle$, pues $\Phi(G)$ consta precisamente de los no generadores de G. Es un resultado conocido (véase [12, III-1.11]) que

$$Y_2 = Y_2(G) = \langle [s_1, s], Y_3(G)\rangle = \langle s_2, Y_3\rangle.$$

Supongamos demostrado que $Y_{i-1} = \langle Y_i, s_{i-1}\rangle$. Como $G = \langle s, s_1\rangle$, se tiene:

$$Y_i = [Y_{i-1}, G] = \langle Y_{i+1}, [s_{i-1}, s], [s_{i-1}, s_1]\rangle.$$

Como G no es excepcional, se tiene que $[s_{i-1}, s_1] \in [Y_{i-1}, Y_1] \leq Y_{i+1}$ para $i \leq m-1$. Además, $[s_{i-1}, s] = s_i$, luego concluimos el resultado deseado. □

Lema 2.1.8. *Sea G un p-grupo de clase maximal y orden p^m con $m \geq 5$. Supongamos que $[Y_i, Y_j] \leq Y_{i+j+1}$ para i, j cualesquiera con $2 < i+j \leq m-2$. Sean s y s_1 elementos de G tales que $G = \langle Y_1, s\rangle$ y $Y_1 = \langle Y_2, s_1\rangle$. Definimos recursivamente los elementos s_i, $2 \leq i \leq m-2$, mediante $s_i = [s_{i-1}, s]$. Entonces se satisfacen las igualdades siguientes:*

$$[s_1, s_{m-2}] = [s_2, s_{m-3}]^{-1} = \cdots = [s_i, s_{m-i-1}]^{(-1)^{i-1}} = \cdots = [s_{m-2}, s_1]^{(-1)^{m-1}}.$$

Demostración. Según nuestra hipótesis, tenemos que, para i, j cualesquiera con $2 < i + j \leq m - 2$,

$$[Y_i, Y_j] \leq Y_{i+j+1} \tag{2.2}$$

Supongamos que $2 \leq i \leq m - 2$. Entonces tenemos:

$$[s_i, s_{m-i-1}] = s_i^{-1} s_i^{s_{m-i-1}} = s_i^{-1}[s_{i-1}, s]^{s_{m-i-1}} = s_i^{-1}[s_{i-1}^{s_{m-i-1}}, s^{s_{m-i-1}}].$$

Por (2.2) tenemos:

$$[s_{i-1}, s_{m-i-1}] \in [Y_{i-1}, Y_{m-i-1}] \leq Y_{m-1} = Z(G).$$

Así, pues, tenemos:

$$\begin{aligned}
[s_i, s_{m-i-1}] &= s_i^{-1} s_i^{s_{m-i-1}} \\
&= s_i^{-1}[s_{i-1}, s]^{s_{m-i-1}} \\
&= s_i^{-1}[s_{i-1}^{s_{m-i-1}}, s^{s_{m-i-1}}] \\
&= s_i^{-1}\big[s_{i-1}[s_{i-1}, s_{m-i-1}], s[s, s_{m-i-1}]\big] \\
&= s_i^{-1}\big[s_{i-1}, s[s, s_{m-i-1}]\big] \\
&= s_i^{-1}[s_{i-1}, s s_{m-i}^{-1}] \\
&= s_i^{-1}[s_{i-1}, s_{m-i}^{-1}][s_{i-1}, s]^{s_{m-i}^{-1}},
\end{aligned}$$

utilizando propiedades de los conmutadores.
Se verifica, además, que

$$[s_{i-1}, s_{m-i}^{-1}] \in [Y_{i-1}, Y_{m-i}] \leq Y_{m-1} = Z(G),$$

y, por otro lado, $[s_{i-1}, s] = s_i$, por tanto, se tiene que

$$\begin{aligned}
[s_i, s_{m-i-1}] &= [s_{i-1}, s_{m-i}^{-1}] s_i^{-1} s_i^{s_{m-i}^{-1}} \\
&= [s_{i-1}, s_{m-i}^{-1}][s_i, s_{m-i}^{-1}],
\end{aligned}$$

pero $[s_i, s_{m-1}^{-1}] \in [Y_i, Y_{m-i}] \leq Y_m = 1$, y, teniendo en cuenta que $[s_{i-1}, s_{m-i}^{-1}] = [s_{i-1}, s_{m-i}]^{-1}$, podemos concluir que

$$[s_i, s_{m-i-1}] = [s_{i-1}, s_{m-i}]^{-1}$$

para $2 \leq i \leq m - 2$, como queríamos probar. □

Lema 2.1.9. *Sea G un p-grupo de clase maximal y orden p^m, con $m \geq 5$. Entonces las siguientes afirmaciones son equivalentes dos a dos:*

1. *G no es un p-grupo excepcional, esto es, $[Y_i, Y_1] \leq Y_{i+2}$ para $2 \leq i \leq m-2$.*
2. *Se verifica que $[Y_i, Y_j] \leq Y_{i+j+1}$ para i, j cualesquiera con $i + j > 2$, conviniendo que $Y_k = 1$ si $k \geq m$.*
3. *Los grupos cocientes abelianos Y_i/Y_{i+2} (con $2 \leq i \leq m-2$) de orden p^2 son isomorfos dos a dos como G-grupos.*

Demostración. Procedemos por inducción sobre $|G|$. Para ver que 1 implica 2, tenemos en cuenta que, como G no es excepcional, se verifica que

$$[Y_1, Y_i] \leq Y_{i+2} \qquad \text{para todo } i \geq 2. \tag{2.3}$$

Por tanto, G/Y_{m-1} no es excepcional y, por la hipótesis inductiva aplicada a G/Y_{m-1}, tenemos que

$$[Y_i, Y_j] \leq Y_{i+j+1} \qquad \text{para } 2 < i + j \leq m-2. \tag{2.4}$$

Por tanto, las hipótesis del Lema 2.1.8 se satisfacen. Por (2.3), tenemos:

$$[s_1, s_{m-2}] \in [Y_1, Y_{m-2}] \leq Y_m = 1,$$

y, usando el Lema 2.1.8, se obtiene que, para $i \leq m-2$, $1 = [s_1, s_{m-2}] = [s_i, s_{m-i-1}]$. Según el Lema 2.1.7, se tiene que $Y_i = \langle Y_{i+1}, s_i \rangle$ y que $Y_{m-i-1} = \langle Y_{m-1}, s_{m-i-1} \rangle$. Además, se tiene:

$$\begin{gathered}[Y_{m-i-1}, Y_{i+1}] \leq Y_m = 1,\\ [s_{m-i-1}, s_i] = 1,\\ [Y_{m-i}, s_i] \leq Y_m = 1,\end{gathered}$$

por tanto, $[Y_i, Y_{m-i-1}] = 1$, es decir, $[Y_i, Y_j] = 1$ para $i + j = m-1$.
Para $i + j > m$ con i, $j \geq 2$, se sigue 2, ya que tenemos que

$$[Y_i, Y_j] \leq Y_{i+j} \leq Y_m = 1,$$

según propiedades muy conocidas de los conmutadores (véase [12, III-2.11]). Por tanto, queda demostrada esta implicación.

Para ver que 2 implica 3, consideramos $z_i \in G \setminus C_G(Y_i/Y_{i+2})$ para $2 \leq i \leq m-2$. La aplicación τ_i definida por

$$(xY_{i+2})^{\tau_i} = [x, z_i]Y_{i+3}, \quad x \in Y_i,$$

es un homomorfismo de Y_i/Y_{i+2} en Y_{i+1}/Y_{i+3} para $2 \leq i \leq m-3$. En efecto, para la demostración podemos suponer que $Y_{i+3} = 1$. Como $[Y_{i+2}, z_i] = 1$, τ_i es efectivamente una aplicación. Para x_1, $x_2 \in G$ cualesquiera, teniendo en cuenta que $[x_1, z_i, x_2] \in [Y_i, G, Y_1] = 1$, tenemos que

$$\begin{aligned}(x_1x_2Y_{i+2})^{\tau_i} &= [x_1x_2, z_i] \\ &= [x_1, z_i][x_1, z_i, x_2][x_2, z_i] \\ &= [x_1, z_1][x_2, z_i] \\ &= (x_1Y_{i+2})^{\tau_i}(x_2Y_{i+2})^{\tau_i}.\end{aligned}$$

Además, τ_i es un G-homomorfismo. En efecto, para $x \in Y_i$ y $g \in G$, debe ser $[x^g, z_i] = [x^g, z_i^g]$. Por tanto, es suficiente demostrar que $z_i^g z_i^{-1} \in C_G(Y_i)$. Ahora bien, por la hipótesis 2, se tiene que

$$[x, z_i^g z_i^{-1}] \in [Y_1, Y_2] \leq Y_{i+3} = 1 \qquad \text{para } x \in Y_i \text{ y para } i \leq m-3.$$

Como Y_i es un G-homomorfismo de Y_i/Y_{i+2} en Y_{i+1}, se tiene que $\operatorname{Im}\tau_i$ y $\operatorname{Ker}\tau_i$ son subgrupos normales de G. Como $z_i \notin C_G(Y_i/Y_{i+2})$, se tiene que $\operatorname{Im}\tau_i \not\leq Y_{i+2}$, así, pues, se tiene que $\operatorname{Im}\tau_i = Y_{i+1}$, y, además, $\operatorname{Ker}\tau_i = Y_{i+2}$. Por tanto, concluimos que τ_i es un G-isomorfismo de Y_i/Y_{i+2} sobre Y_{i+1}/Y_{i+3}.

Por último, para ver que 3 implica 1, supongamos que los grupos de orden p^2 Y_i/Y_{i+2}, con $2 \leq i \leq m-2$, son todos G-isomorfos. Entonces todos los centralizadores son iguales, a saber, iguales a Y_1, luego G no es un p-grupo excepcional. □

A continuación veremos que los subgrupos maximales de G distintos de Y_1 de un p-grupo de clase maximal no excepcional son p-grupos de clase maximal. Esta demostración se hará por medio de varios lemas.

El Teorema 2.1.12 es un resultado de **P. Hall** y aparece demostrado en [12, III-7.10]. Necesitaremos los siguientes resultados previos. El primero de ellos aparece en [12, III-2.11].

Teorema 2.1.10 (Hall). *1. Para cada serie central*

$$N_1 \geq \cdots \geq N_r$$

de G, se tiene que $[N_i, Y_j(G)] \leq N_{i+j}$, donde convenimos que $N_k = N_r$ para $k > r$.

2. *Se tiene que $[Y_i(G), Y_j(G)] \leq Y_{i+j}(G)$ para i, $j \geq 2$. Por consiguiente, si G es nilpotente de clase c, $Y_i(G)$ es abeliano para $2i > c$.*

3. *Se da la inclusión*

$$[Y_i(G), Z_j(G)] \leq Z_{j-i}(G),$$

donde para $j \leq i$ entendemos que $Z_{j-i}(G) = 1$. En particular, se tiene que $[Y_i(G), Z_i(G)] = 1$, con lo que cada elemento de $Y_i(G)$ conmuta con cada elemento de $Z_i(G)$.

Demostración. 1. Como los N_i forman una serie central, se tiene la inclusión $[N_i, G] \leq N_{i+1}$. Tenemos entonces que

$$\big[[N_i, G], G\big] \leq [N_{i+1}, G] \leq N_{i+2}, \tag{2.5}$$
$$\big[[G, N_i], G\big] \leq [N_{i+1}, G] \leq N_{i+2}, \tag{2.6}$$

de donde se deduce que

$$\big[[G, G], N_i\big] \leq N_{i+2},$$

según se deduce de [12, III-2.8].

Razonemos ahora por inducción. Supongamos que $[N_i, Y_j] \leq N_{i+j}$ para $j \geq 2$. Se tiene entonces que

$$\big[[N_i, Y_1], G\big] \leq [N_{i+1}, G] \leq N_{i+j+1} \tag{2.7}$$
$$\big[[G, N_i], Y_j\big] \leq [N_{i+1}, Y_j] \leq N_{i+j+1}, \tag{2.8}$$

de donde se obtiene que

$$\big[[Y_j, G], N_i\big] = [Y_{j+1}, N_i] = [N_i, Y_{j+1}] \leq N_{i+j+1},$$

como queríamos probar.

2. Con lo anterior aplicado a $N_i = Y_i(G)$, y si tenemos en cuenta que

$$\big(Y_i(G)\big)' = [Y_i(G), Y_i(G)] \leq Y_{2i}(G),$$

concluimos que si $2i > c$, entonces $Y_i(G)$ es abeliano.

3. Si $N_i = Z_{j-i-1}(G)$, se tiene que

$$[Y_i(G), Z_j(G)] = [Y_i(G), N_1] \leq N_{i+1} = Z_{j-i}(G). \quad \square$$

El siguiente resultado aparece en [12, III-7.10].

Lema 2.1.11. *Sea N un subgrupo normal no abeliano de G con $N \leq Y_i(G)$. Se tienen entonces las siguientes relaciones:*

$$|Z(N)| \geq p^i, \qquad |N| \geq p^{i+2}, \qquad |N/N'| \geq p^{i+1}.$$

Demostración. Se tiene que $[Y_i(G), Z_i(G)] = 1$, según hemos visto en el teorema anterior. De este modo, se sigue que $N \not\leq Z_i(G)$, pues, en otro caso, N sería abeliano. De este modo, $N \cap Z_i(G) \leq Z(N)$. Para $j \leq i$, tenemos:

$$\begin{aligned} 1 &< Z\big(G/Z_j(G)\big) \cap NZ_j(G)/Z_j(G) \\ &= Z_{j+1}(G)/Z_j(G) \cap NZ_j(G)/Z_j(G) \\ &= \big(Z_{j+1}(G) \cap N\big)Z_j(G)/Z_j(G). \end{aligned} \tag{2.9}$$

De este modo, se tiene que

$$1 < N \cap Z_1(G) < N \cap Z_2(G) < \ldots < N \cap N_i(G),$$

pues en otro caso se contradiría (2.9), y podemos deducir que

$$|Z(N)| \geq |N \cap Z_i(G)| \geq p^i.$$

Teniendo en cuenta que $|N/Z(N)| \geq p^2$, por ser N no abeliano, concluimos que $|N| \geq p^{i+2}$.

Sea M un subgrupo normal de G tal que $|N'/M| = p$. Escribamos $\overline{N} = N/M$. Entonces tenemos que

$$\overline{N} \trianglelefteq G/M,$$

con $\overline{N}' = N'/M \neq 1$ y $\overline{N} \leq Y_i(G)M/M = Y_i(G/M)$. Como $|\overline{N}| \geq p^{i+2}$, se sigue que $|N/N'| = |\overline{N}/\overline{N}'| \geq p^{i+1}$. $\square$

Lema 2.1.12 (Hall)**.** *Supongamos que $G^{(k)} \neq 1$. Entonces $|G| \geq p^{2^k+k}$.*

Demostración. Se tiene que $1 < G^{(i+1)} < G^{(i)}$. Por una propiedad conocida de conmutadores, se tiene que $G^{(i)} \leq Y_{2^i}(G)$. Por el Lema 2.1.11, se tiene que

$$|G^{(i)}/G^{(i+1)}| \geq p^{2^i+1}.$$

Si $|G| = p^n$, teniendo en cuenta que $|G^{(k)}| \geq p$, resulta que

$$m \geq 1 + \sum_{i=0}^{k-1}(2^i + 1) = 2^k + k,$$

como queríamos probar. □

Teorema 2.1.13. *Sea G un p-grupo de clase maximal. Si $G'' = 1$, entonces G no es excepcional.*

Demostración. Haremos la demostración por inducción sobre $|G| = p^m$. Por tanto, podemos suponer que $m \geq 5$ y que G/Y_{m-1} no es excepcional. De este modo, tenemos que, para $2 \leq i \leq m-3$,

$$[Y_i, Y_i] \leq Y_{i+2}.$$

De 2.1.8 se sigue que $[s_1, s_{m-2}] = [s_2, s_{m-3}]^{-1} \in [Y_2, Y_2] = 1$. Por tanto, se obtiene que $Y_1 = \langle s_1, Y_2\rangle \leq C_G(Y_{m-2})$, luego $Y_1 = C_G(Y_{m-2})$ y, en consecuencia, G no es excepcional. □

Teorema 2.1.14. *Sea G un p-grupo de clase maximal y orden p^m, con $5 \leq m \leq p+2$. Entonces se tiene:*

1. *G/Y_{m-1} no es un grupo excepcional, así, pues, $Y_1 = C_G(Y_1/Y_{i+2})$ para $2 \leq i \leq m-3$.*

2. *Si G es un grupo excepcional, entonces $p > 3$, m es par y también $6 \leq m \leq p+1$.*

Demostración. Razonaremos por inducción sobre m.

Si $m = 5$, entonces, por 2.1.12, se tiene que $G'' = 1$ y, por 2.1.13, G no es excepcional.

Supongamos ahora que $6 \leq m \leq p+2$.

Etapa 1 Supongamos que G/Y_{m-1} no es excepcional. Según 2.1.9, se verifica la siguiente relación para $2 < i + j \leq m - 2$:

$$[Y_i, Y_j] \leq Y_{i+j+1}.$$

Además, de 2.1.8 se sigue que $[s_1, s_{m-2}] = [s_1, s_{m-2}]^{(-1)^m}$. Si m es impar, como p es mayor que 2, se sigue que $[s_1, s_{m-2}] = 1$ y, entonces, también se tiene que $Y_1 = C_G(Y_i/Y_{i+2})$ para $2 \leq i \leq m - 2$ y, por consiguiente, G no es excepcional.

Etapa 2 Supongamos que G/Y_{m-1} es un grupo excepcional. Como $m \geq 6$, podemos aplicar la hipótesis inductiva a G/Y_{m-2}, se sigue entonces que G/Y_{m-2} no es excepcional y m es impar. Elegimos elementos distintos s y s_1 en G de manera que $G = \langle Y_1, s\rangle$ y que $Y_1 = \langle Y_2, s_1\rangle$, y definimos los elementos s_i de Y_i como en reflema:14.8 por la regla

$$s_i = [s_{i-1}, s]. \tag{2.10}$$

Aplicando 2.1.9 a G/Y_{m-2}, obtenemos que para $2 < i + j \leq m - 3$, se obtiene la siguiente relación:

$$[s_i, s_j] \in [Y_i, Y_j] \leq Y_{i+j+1}. \tag{2.11}$$

Como G/Y_{m-2} no es excepcional, podemos aplicar 2.1.8 a G/Y_{m-1}, con lo que se obtiene lo siguiente para $2 \leq i \leq m - 3$:

$$[s_1, s_{m-3}] \equiv [s_i, s_{m-i-2}]^{(-1)^{i-1}} \quad (\text{mód } Y_{m-1}). \tag{2.12}$$

Como G/Y_{m-1} es excepcional y G/Y_{m-2} no lo es, se verifica la inclusión $[Y_1, Y_i] \leq Y_{i+2}$ para $i \leq m - 4$ y $[Y_1, Y_{m-3}] \not\leq Y_{m-1}$, pero $[Y_1, Y_{m-3}] \leq Y_{m-2}$. Por ser Y_{m-2} el único subgrupo normal de G de orden p, se sigue que $[Y_1, Y_{m-3}] = Y_{m-2}$. Ponemos entonces

$$s_{m-2} = [s_{m-3}, s_1]. \tag{2.13}$$

Se tiene que $Y_1 = \langle Y_2, s_1\rangle$. Como G/Y_{m-2} no es excepcional, según 2.1.7 se verifica que $Y_{m-3} = \langle Y_{m-2}, s_{m-3}\rangle$, y, consecuentemente,

$$Y_{m-2} = [Y_1, Y_{m-3}] = \langle Y_{m-1}, s_{m-2}\rangle. \tag{2.14}$$

De (2.12) y (2.13) obtenemos:

$$[s_i, s_{m-i-2}] \equiv s_{m-2}^{(-1)^i} \quad (\text{mód } Y_{m-1}), \qquad i \in \{2, \ldots, m-3\}. \tag{2.15}$$

En particular, tenemos que $[s_2, s_{m-4}] \equiv s_{m-2}$ (mód Y_{m-1}), y como, además, $Y_{m-1} = Z(G)$, se tiene:

$$[s_{m-2}, s_1] = \big[[s_2, s_{m-4}], s_1\big] = [s_2, s_{m-4}]^{-1}[s_2, s_{m-4}]^{s_1}. \tag{2.16}$$

Por tanto, tenemos:

$$[s_2, s_{m-4}]^{s_1} = [s_2^{s_1}, s_{m-4}^{s_1}] = \big[s_2[s_2, s_1], s_{m-4}[s_{m-4}, s_1]\big].$$

Como $m > 5$, de (2.11) se tiene que $[s_2, s_1] \in Y_4$. Por tanto, $[s_2, s_1]$ conmuta con el elemento $s_{m-4}^{s_1}$ de Y_{m-4}. De este modo,

$$[s_2, s_{m-4}]^{s_1} = \big[s_2, s_{m-4}[s_{m-4}, s_1]\big]^{[s_2, s_1]}.$$

Como $[s_2, s_1] \in Y_4$, $[s_2, Y_{m-4}] \leq Y_{m-2}$ y $[Y_4, Y_{m-2}] = 1$, se sigue que

$$[s_2, s_{m-4}]^{s_1} = \big[s_2, s_{m-4}[s_{m-4}, s_1]\big].$$

De (2.11) se obtiene que $[s_{m-4}, s_1] \in Y_{m-2}$. Como $[Y_2, Y_{m-2}] = 1$, se sigue que

$$[s_2, s_{m-4}]^{s_1} = [s_2, s_{m-4}]^{[s_{m-4}, s_1]} = [s_2, s_{m-4}].$$

Ahora bien, de (2.16) se sigue que $[s_{m-2}, s_1] = 1$, luego $s_{m-2} \in C_G(Y_1)$, pues $Y_1 = \langle Y_2, s_1\rangle$ y $[s_{m-2}, Y_2] = 1$. Por (2.14) tenemos que $Y_{m-2} = \langle Y_{m-1}, s_{m-2}\rangle = C_G(Y_1)$. Así, pues,

$$[Y_1, Y_{m-2}] = 1. \tag{2.17}$$

Como $Y_{m-2} \neq Y_{m-1} = Z(G)$, es claro que $s_{m-2} \notin Z(G)$. De este modo, $C_G(Y_{m-2}) = C_G(s_{m-2}) = Y_1$ y $[s_{m-2}, s] \neq 1$. Pongamos

$$s_{m-1} = [s_{m-2}, s]. \tag{2.18}$$

Entonces se verifica que $Y_{m-1} = \langle s_{m-1}\rangle$, pues Y_{m-1} es cíclico de orden p.

Demostraremos ahora por inducción sobre i que

$$[s_i, s_{m-i-1}] = s_{m-1}^{(-1)^{i-1}(i-1)} \qquad \text{para } i \in \{2, \ldots, m-3\}. \tag{2.19}$$

Para $i = 2$, se tiene que

$$\begin{aligned}[s_2, s_{m-3}] &= s_2^{-1} s_2^{s_{m-3}} \\ &= s_2^{-1}[s_1, s]^{s_{m-3}} \\ &= s_2^{-1}[s_1^{s_{m-3}}, s^{s_{m-3}}] \\ &= s_2^{-1}\big[s_1[s_1, s_{m-3}], s[s, s_{m-3}]\big].\end{aligned}$$

Como $[s, s_{m-3}] \in Y_{m-2}$, de (2.17) se sigue que $[s, s_{m-3}]$ conmuta con cada elemento de Y_1. Así, pues, por (2.13) tenemos:

$$[s_2, s_{m-3}] = s_2^{-1}[s_1 s_{m-2}^{-1}, s] = s_2^{-1}[s_1, s]^{s_{m-2}^{-1}}[s_{m-2}^{-1}, s].$$

Dado que $[s_1, s] = s_2 \in C_G(Y_{m-2})$ y que $[s_{m-2}, s] \in Y_{m-1} = Z(G)$, se sigue que

$$[s_2, s_{m-3}] = s_2^{-1} s_2 [s_{m-2}, s]^{-1} = s_{m-1}^{-1},$$

que es la afirmación (2.19) para $i = 2$.

Supongamos que $i > 2$ y que está demostrado que

$$[s_{i-1}, s_{m-i}] = s_{m-1}^{(-1)^i (i-2)}. \tag{2.20}$$

Como $2 < i \leq m - 3$, por (2.10) se sigue que

$$\begin{aligned}[s_i, s_{m-i-1}] &= s_i^{-1}[s_{i-1}, s]^{s_{m-i-1}} \\ &= s_i^{-1}[s_{i-1}^{s_{m-i-1}}, s^{s_{m-i-1}}] \\ &= s_i^{-1}\big[s_{i-1}[s_{i-1}, s_{m-i-1}], s[s, s_{m-i-1}]\big] \\ &= s_i^{-1}\big[s_{i-1}[s_{i-1}, s_{m-i-1}], s s_{m-i}^{-1}\big].\end{aligned}$$

Según (2.12), con $i - 1$ en lugar de i, y (2.13), se tiene que

$$[s_{i-1}, s_{m-i-1}] \equiv [s_1, s_{m-3}]^{(-1)^i} \equiv s_{m-2}^{(-1)^{i-1}} \pmod{Y_{m-1}}.$$

Como $Y_{m-1} = Z(G)$ se sigue que

$$\begin{aligned}[s_i, s_{m-i-1}] &= s_i^{-1}[s_{i-1} s_{m-2}^{(-1)^{i-1}}, s s_{m-i}^{-1}] \\ &= s_{i-1}[s_i^{-1} s_{m-2}^{(-1)^{i-1}}, s_{m-i}^{-1}][s_{i-1} s_{m-2}^{(-1)^{i-1}}, s]^{s_{m-i}^{-1}} \\ &= s_i^{-1}[s_{i-1} s_{m-2}^{(-1)^{i-1}}, s_{m-i}^{-1}][s_{i-1} s_{m-2}^{(-1)^{i-1}}, s],\end{aligned}$$

pues $[Y_i, s_{m-i}] \leq Y_m = 1$. Tenemos que $[s_{i-1}, s_{m-i}^{-1}] = [s_{i-1}, s_{m-i}]^{-1} \in Y_{m-1} = Z(G)$, $[s_{m-2}^{(-1)^{i-1}}, s_{m-i}^{-1}] \in [Y_{m-2}, Y_2] = 1$ (pues $i \leq m-3$), $[s_{i-1}, s] = s_i$ y $[s_{m-2}^{(-1)^{i-1}}, s] = s_{m-1}^{(-1)^{i-1}}$ (usando (2.18)). De este modo, por (2.20) se sigue que

$$\begin{aligned}[s_i, s_{m-i-1}] &= s_i^{-1}[s_{i-1}, s_{m-i}]^{-1} s_i s_{m-1}^{(-1)^{i-1}} \\ &= s_{m-1}^{(-1)^{i-1}(i-2)} s_{m-1}^{(-1)^{i-1}} \\ &= s_{m-1}^{(-1)^{i-1}(i-1)},\end{aligned}$$

y, consecuentemente, (2.19) queda demostrado. Ahora ponemos en (2.19) $i = (m-1)/2$ (esto tiene sentido, pues m es impar y también, como $m \geq 5$, $(m-1)/2 \leq m-3$), y obtenemos que

$$s_{m-1}^{(m-3)/2} = 1.$$

Como $s_{m-1} \neq 1$, se tiene que $m - 3 \equiv 0 \pmod{p}$, en contra de nuestra hipótesis $m \leq p+2$. De este modo, se tiene nuestro resultado para $m \leq p+2$. □

Lema 2.1.15. *Sea G un p-grupo de clase maximal y orden p^m, con $m \geq 4$. Sea $s \in G$ y $s \notin C_G(Y_i/Y_{i+2})$ para cada $i \in \{2, \ldots, m-2\}$ (en particular, $s \notin C_G(Y_2/Y_4) = Y_1$ y, por tanto, $G = \langle Y_1, s\rangle$). Entonces se tiene:*

1. $C_G(s) = \langle Y_{m-1}, s\rangle = \langle s\rangle Y_{m-1}$.
2. $s^p \in Y_{m-1}$, *por tanto,* $o(s) \leq p^2$ *y* $|C_G(s)| = p^2$.
3. $\mathrm{Cl}_G(s) = sY_2$.
4. *Si* $sY_2 = s'Y_2$, *entonces* $s^p = (s')^p$.

Demostración. 1. Tenemos que $G = \langle Y_1, s\rangle = \langle s\rangle Y_1$, luego si $g \in G$, entonces se tiene que $g = s^j x$ para algún $x \in Y_1$. Por tanto, se sigue directamente el resultado deseado si probamos que $Y_1 \cap C_G(s) = Y_{m-1}$. Consideremos $z \in Y_1 \cap C_G(s)$. Entonces existe $j \geq 1$ tal que $z \in Y_j \backslash Y_{j+1}$. Supongamos que $j = 1$. Entonces $Y_1 = \langle Y_2, z\rangle$. Además, $G = \langle Y_1, s\rangle$ y $z \in C_G(s)$, luego $G = \langle Y_2, z, s\rangle = \langle \Phi(G), z, s\rangle = \langle z, s\rangle$, abeliano, lo cual es imposible. Supongamos que $2 \leq j \leq m-2$. Entonces $Y_j = \langle Y_{j+1}, z\rangle$, $[s, z] = 1$ y $[s, Y_{j+1}] \leq Y_{j+2}$. Esto origina que $[s, Y_j] \leq Y_{j+2}$, en contra de la elección de s. Así, pues, $j = m-1$ y $z \in Y_{m-1}$, esto es, $Y_1 \cap C_G(s) = Y_{m-1}$.

2. Tenemos que $G/Y_1 = \langle \bar{s} \rangle \cong C_p$, luego $s^p \in Y_1 \cap C_G(s) = Y_{m-1}$, por tanto, $o(s) \leq p^2$. Además, $s \notin Y_{m-1}$, luego $|C_G(s)| = |\langle s \rangle Y_{m-1}| = p|Y_{m-1}| = p^2$.

3. Según el apartado anterior, tenemos que $|\mathrm{Cl}_G(s)| = p^{m-2}$ y $\mathrm{Cl}_G(s) \subseteq sY_2$, pues $s^g = s[s,g] \in sY_2$ y también $|sY_2| = |Y_2| = p^{m-2}$, luego necesariamente $\mathrm{Cl}_G(s) = sY_2$.

4. Si $sY_2 = s'Y_2$, según el apartado anterior existe $x \in G$ tal que $s' = s^x$ y, según 2, $s^p \in Z(G)$. Por tanto,

$$(s')^p = (s^x)^p = (s^p)^x = s^p,$$

como queríamos probar. □

Nota 2.1.16. Si hubiéramos demostrado que G/Y_1 no es excepcional, entonces tendríamos que $Y_1 = C_G(Y_1/Y_{i+2})$ para $2 \leq i \leq m-3$, y como G tiene exactamente $p+1$ subgrupos maximales de índice p (pues $G/G' \cong C_p \times C_p$), se seguiría que existiría $U \leq G$ tal que $|G:U| = p$ y $Y_1 \neq U \neq C_G(Y_{m-2})$. Si elegimos s tal que $U = \langle Y_2, s \rangle$, entonces s tiene la propiedad dada en 2.1.15.

Para los siguientes resultados necesitaremos hacer uso de la noción de p-grupo regular. Los resultados sobre p-grupos regulares aparecen demostrados en [12, III-10].

Definición 2.1.17. Dado un p-grupo G, definimos los siguientes subgrupos:

$$\Omega_i(G) = \langle g \in G \mid g^{p^i} = 1 \rangle$$

y

$$\mho_i(G) = \langle g^{p^i} \mid g \in G \rangle.$$

Se denota $p^{\omega(G)} = |G/\mho_1(G)|$.
Un p-grupo G se dice *regular* si, para cada $x \in G$, $y \in G$, se verifica que

$$x^p y^p = (xy)^p \prod_i d_i^p$$

con d_i elementos de $\langle x, y \rangle'$.

Es sencillo observar que los subgrupos $\Omega_i(G)$ y $\mho_i(G)$ son característicos.
El siguiente teorema aparece en [12, Satz III-10.2].

Teorema 2.1.18. *Sea G un p-grupo. Se tiene:*

1. *Si la clase de nilpotencia de G es a lo sumo p, entonces G es regular.*
2. *Si $|G| \leq p^p$, entonces G es regular.*
3. *Si G' es cíclico y $p > 2$, entonces G es regular.*
4. *Si $\exp G = p$, entonces G es regular.*

El siguiente teorema está en [12, Haupsatz III-10.5].

Teorema 2.1.19. *Sea G un p-grupo regular, y k un número natural.*

1. *Si $x^{p^k} = y^{p^k} = 1$, entonces $(xy)^{p^k} = 1$. Los elementos x de G tales que $x^{p^k} = 1$ componen el subgrupo característico de G $\Omega_k(G)$.*
2. *Para cada x, $y \in G$, $x^{p^k} y^{p^k} = z^{p^k}$ para un elemento adecuado $z \in G$, que dependerá de x y de y. Las potencias p^k-ésimas de elementos de G componen el subgrupo característico $\mathfrak{V}_k(G)$ de G.*

El siguiente resultado aparece en [12, Satz III-10.7].

Teorema 2.1.20. *Sea G un p-grupo regular.*

1. *Se tiene que $|G/\Omega_k(G)| = |\mathfrak{V}_k(G)|$.*
2. *Escribamos $|\Omega_k(G)/\Omega_{k-1}(G)| = p^{\omega_k}$, entonces se tiene que*

$$\omega_1 \geq \omega_2 \geq \cdots \geq \omega_\mu,$$

donde p^μ es el exponente de G.

Por último, el siguiente resultado aparece en [12, Satz III-10.13].

Teorema 2.1.21. *Si se tiene que $\omega\big(Y_i(G)\big) \leq p-i$ para algún i con $2 \leq i \leq p$ o si $\omega(G) \leq p-1$, , entonces G es regular.*

Lema 2.1.22. *Sea G un p-grupo de clase maximal y orden p^m con $5 \leq m \leq p+1$. Entonces Y_2 y G/Y_{m-1} tienen ambos exponente p.*

Demostración. De 2.1.14, se tiene que G/Y_{m-1} no es excepcional. Por tanto, se verifica que

$$Y_1 = C_G(Y_1/Y_{i+2})$$

para $2 \leq i \leq m-3$, aunque puede ser $Y_1 \neq C_G(Y_{m-2})$. Como $p > 2$ y G tiene exactamente $p+1 \geq 4$ subgrupos maximales, entonces existen dos de ellos $\langle s, Y_2\rangle$ y $\langle s', Y_2\rangle$ distintos de todos los $C_G(Y_i/Y_{i+2})$, $2 \leq i \leq m-2$. Así, pues, tenemos que s, $s' \notin C_G(Y_i/Y_{i+2})$ para $2 \leq i \leq m-2$. Se tiene entonces que $G = \langle s, s', Y_2\rangle = \langle s, s', \Phi(G)\rangle = \langle s, s'\rangle$ y, además, se satisfacen las hipótesis de 2.1.15, luego $s^p \in Y_{m-2}$ y $(s')^p \in Y_{m-1}$. Por tanto, Y_{m-1} está generado por dos elementos de orden a lo sumo p. Como $|G/Y_{m-1}| = p^{m-1} \leq p^p$, según 2.1.18, G es regular y, consecuentemente, según 2.1.19, G tiene exponente p. Como $|Y_1| = p^{m-1} \leq p^p$, usando de nuevo 2.1.18, Y_1 también es regular. Se sigue entonces que $\mho_1(Y_1) \leq \mho_1(G) \leq Y_{m-1}$. Así, pues, por 2.1.20, $|\Omega_1(Y_1)| = |Y_1/\mho_1(Y_1)| \geq |Y_1/Y_{m-1}| = p^{m-2}$. Por tanto, $Y_2 \leq \Omega_1(Y_1)$ y, de este modo, Y_2 tiene exponente p. □

Lema 2.1.23. *Sea G un p-grupo de clase maximal y orden p^m con $m > p+1$. Sea $Y_1 = \langle Y_2, s_1\rangle$. Entonces se tiene que $s_1^p \in Y_p \setminus Y_{p+1}$.*

Demostración. Aplicando 2.1.22 a G/Y_{p+1} obtenemos que $s_1^p \in Y_p$. Sea $G = \langle Y_1, s\rangle$ y $s_i = [s_{i-1}, s]$ para $i \in \{2, \ldots, p\}$. Como $p+2$ es impar (puesto que $p \neq 2$), G/Y_{p+2} no es excepcional según 2.1.14. Por 2.1.7, se tiene que $Y_p = \langle Y_{p+1}, s_p\rangle$ y $s_p \notin Y_{p+1}$. La identidad de Zassenhaus dice que

$$s_p = [s_1, \overbrace{s, \ldots, s}^{(p-1)}] \in Y_{p+1}\big(\mho_1(G)\big).$$

Como $s_p \notin Y_{p+1}$, se tiene que $\mho_1(G) \not\leq Y_{p+1}$. Por tanto, existe $x \in G$ tal que $x^p \notin Y_{p+1}$. Supongamos que $x \notin Y_1$. Como G/Y_{p+2} no es excepcional y $x \notin Y_1$, se sigue de 2.1.15 la contradicción $x^p \in Y_{p+1}$. Así, pues, $x \in Y_1 = \langle Y_2, s_1\rangle$. Según 2.1.22, tenemos que $\mho_1(Y_2) \leq Y_{p+1}$. Si $s_1^p \in Y_{p+1}$, de la regularidad de Y_1/Y_{p+1} se seguiría que Y_1/Y_{p+1} tiene exponente p. Como $x \in Y_1$, esto origina la contradicción $x^p \in Y_{p+1}$. De este modo, $s_1^p \notin Y_{p+1}$, como queríamos probar. □

Nota 2.1.24. La condición $m > p+1$ es necesaria, como lo prueba el producto orlado regular de dos grupos cíclicos de orden primo p.

Lema 2.1.25. *Sea G un p-grupo de clase maximal y orden p^m, con $m > p+1$. Entonces se verifica que $\mho_1(Y_i) = Y_{i+p-1}$ para $1 \leq i \leq m-p+1$. Además, Y_1 es un p-grupo regular con $\mho_1(Y_1) = Y_{m-p-1}$ y $|Y_1/\mho_1(Y_1)| = p^{p-1}$.*

Demostración. Según 2.1.3, se tiene que $\mho_1(Y_1) = Y_k$ para algún k. La condición del Lema 2.1.22 sobre G/Y_{p+1} origina que $\mho_1(Y_1) \leq Y_p$. Así, pues, $p \leq k$. Si fuese $p < k$, entonces tendríamos que $s_1^p \in \mho_1(Y_1) \leq Y_{p+1}$, en contra de 2.1.23. Por tanto, $\mho_1(Y_1) = Y_p$. De esto se sigue que

$$|Y_1/\mho_1(Y_1)| = |Y_1/Y_p| = p^{p-1},$$

y la regularidad de Y_1 es consecuencia de 2.1.21. Por tanto, según 2.1.20, se tiene que

$$|\Omega_1(Y_1)| = |Y_1/\mho_1(Y_1)| = p^{p-1},$$

y, por consiguiente, $\Omega_1(Y_1) = Y_{m-p+1}$, según 2.1.3. Supongamos que $1 < i \leq m-p+1$. Como Y_i es un subgrupo de Y_1, Y_i es regular. Teniendo en cuenta que $i \leq m-p+1$, $\Omega_1(Y_i) = Y_{m-p+1} \leq Y_i$, así, pues, $\Omega_1(Y_1) = \Omega_1(Y_i)$ y, por tanto, $|Y_i/\mho_1(Y_i)| = |\Omega_1(Y_i) = p^{p-1}$, de este modo, $|G/\mho_1(Y_i)| = p^{i+p-1}$ y de 2.1.3 se sigue que $\mho_1(Y_i) = Y_{i+p-1}$, como queríamos ver. □

La demostración del siguiente resultado aparece en [12, Satz III-11.4].

Lema 2.1.26 (Huppert)**.** *Sea G un p-grupo con $p > 2$. Entonces G es metacíclico cuando se da la condición $|G/\mho_1(G)| \leq p^2$. Los p-grupos regulares con $|G/\mho_1(G)| = p^2$ son los p-grupos metacíclicos no abelianos.*

El siguiente resultado es un corolario de 2.1.25.

Teorema 2.1.27. *Sea G un 3-grupo de clase maximal. Entonces se tiene que $G'' = 1$ y que Y_1 es metacíclico con clase de nilpotencia menor o igual que 2. En particular, cada 3-grupo de clase maximal es no excepcional, según 2.1.13.*

Demostración. Si $|G| \leq 3^4$, entonces se tiene que $|G'| \leq 3^2$ y $G'' = 1$. Supongamos que $|G| = 3^m$, con $m \geq 5$. De 2.1.25 se tiene que $|Y_1/\mho_1(Y_1)| = 3^2$, por tanto, según 2.1.26, Y_1 es un grupo metacíclico y, por tanto, Y_1' es cíclico. Según 2.1.3, existe k tal que $Y_1' = Y_k$. De 2.1.25 se sigue que $\mho_1(Y_{m-2}) = Y_m = 1$. Por tanto, Y_{m-2} tiene exponente 3 y no es cíclico por ser de orden 3^2. De este modo, $k \geq m-1$. Esto implica que $|Y_1'| \leq 3$ y que $Y_1' \leq Z(Y_1)$. Por tanto,

se tiene que $[x^3, y] = [x, y]^3 = 1$ para todo x, $y \in Y_1$, luego $\mho_1(Y_1) \leq Z(Y_1)$. Cada subgrupo maximal de Y_1 es ahora abeliano, ya que es una extensión cíclica de $\mho_1(Y_1) = \Phi(Y_1)$. En particular, esto se satisface para $Y_2 = G'$, como queríamos ver. □

Teorema 2.1.28. *Sea G un p-grupo de clase maximal y orden p^m, con $m \geq p+2$. Entonces se tiene:*

1. *G/Y_{m-1} no es un grupo excepcional, así, pues, $Y_1 = C_G(Y_1/Y_{i+2})$ para $2 \leq i \leq m-3$.*
2. *Si G es un grupo excepcional, entonces $p > 3$, m es par y también $6 \leq m \leq p+1$.*

Demostración. Supongamos ahora que $m \geq p+2$. Tenemos que probar que si G es un p-grupo de clase maximal de orden p^m, con $m \geq p+2$, entonces G no es excepcional. Probaremos esto argumentando por inducción sobre m.

Para $m = p+2$, el resultado es cierto, según hemos visto en 2.1.14.

Supongamos ahora que $m > p+2$. Tenemos que $m-1 \geq p+2$, por tanto, la inducción aplicada a G/Y_{m-1} nos conduce a que G/Y_{m-1} no es excepcional. Como antes, consideremos $G = \langle Y_1, s\rangle$, $Y_1 = \langle Y_2, s_1\rangle$ y $s_{i+1} = [s_i, s]$ para $1 \leq i \leq m-2$. Según 2.1.7, se tiene que $Y_i = \langle Y_{i+1}, s_i\rangle$ para $i \in \{1, 2, \ldots, m-2\}$. Por el Lema 2.1.23, se tiene que $s_1^p \in Y_p \setminus Y_{p+1}$. Así, pues, existen k, con $0 < k < p$, y $x \in Y_{p+1}$ tales que $s_1^p = x s_p^{-k} \in Y_p = Y_{p+1}\langle s_p\rangle$. En consecuencia,

$$[s_1^p s_p^k, s_{m-p-1}] \in [Y_{p+1}, Y_{m-p-1}] \leq Y_m = 1.$$

Luego se tiene que

$$1 = [s_1^p, s_{m-p-1}]^{s_p^k}[s_p^k, s_{m-p-1}].$$

Como $[s_1^p, s_{m-p-1}] \in [Y_p, Y_{m-p-1}] \leq Y_{m-1} = Z(G)$ y $[s_p, s_{m-p-1}] \in Y_{m-1}$, se sigue que

$$[s_{m-p-1}, s_p]^k = [s_1^p, s_{m-p-1}] = s_1^{-p}(s_1^p)^{s_{m-p-1}} = s_1^{-p}(s_1^{s_{m-p-1}})^p = s_1^{-p}(s_1 y)^p,$$

con $y = [s_1, s_{m-p-1}]$. Como G/Y_{m-1} no es excepcional, y observando que $m-p-1 \leq m-3$, se sigue de 2.1.9 que $y = [s_1, s_{m-p-1}] \in [Y_1, Y_{m-p-1}] \leq Y_{m-p+1}$. Según 2.1.25, Y_1 es regular y $\Omega_1(Y_1) = Y_{m-p-1}$ tiene exponente p. Como $y \in Y_{m-p-1}$, se tiene también que $\langle s_1, y\rangle'$ está contenido en Y_{m-p+1}. El carácter regular de Y_1 origina que $(s_1 y)^p = s_1^p y^p = s_1^p$. Por tanto, $[s_{m-p-1}, s_p]^k = 1$.

Como k no es un múltiplo de p, se sigue que $[s_{m-p-1}, s_p] = 1$. Teniendo en cuenta que G/Y_{m-1} no es excepcional, obtenemos de 2.1.9 y de 2.1.8 que $[s_1, s_{m-2}] = [s_p, s_{m-p-1}]^{(-1)^{p-1}} = 1$. Tenemos que $Y_1 = \langle Y_2, s_1 \rangle$. Aplicando 2.1.7 a G/Y_{m-1} obtenemos que $Y_{m-2} = \langle Y_{m-1}, s_{m-2} \rangle$. Por tanto, $[Y_1, Y_{m-2}] = 1$. Como G/Y_{m-1} no es excepcional, también se verifica que

$$[Y_1, Y_i] \leq Y_{i+2}$$

para $i \leq m-3$. Por tanto, G es, asimismo, no excepcional, como queríamos probar. □

Teorema 2.1.29. *Sea G un p-grupo de clase maximal. Si G no es excepcional, entonces cada uno de los subgrupos maximales de G distintos de Y_1 es un p-grupo de clase maximal.*

Demostración. Tenemos que probar que cada subgrupo maximal H de G distinto de Y_1 tiene clase $m-2$. Sea $G = \langle Y_1, s \rangle$, $Y_1 = \langle Y_2, s_1 \rangle$ y $s_{i+1} = [s_i, s]$. Entonces cada subgrupo maximal H de G distinto de Y_1 tiene la forma $H = \langle Y_2, ss_1^i \rangle$ para un i adecuado con $0 \leq i \leq p-1$, ya que $G/G' = \langle sG', s_1G' \rangle \cong C_p^2$. Demostraremos por inducción sobre j que, para $1 \leq j \leq m-3$,

$$[s_2, \overbrace{ss_1^i, \ldots, ss_1^i}^{(j)}] \equiv s_{j+2} \pmod{Y_{j+3}}$$

Para $j = 1$, tenemos que $[s_2, s_1] \in Y_4$ y, por tanto, se sigue el resultado trabajando módulo Y_4. Supongamos que

$$[s_2, \overbrace{ss_1^i, \ldots, ss_1^i}^{(j-1)}] = s_{j+1}y,$$

con $y \in Y_{j+2}$ y $j \leq m-3$. Entonces se tienen las siguientes congruencias módulo Y_{j+3}:

$$\begin{aligned}
[s_{j+1}y, ss_1^i] &= [s_{j+1}, ss_1^i]^y [y, ss_1^i] \\
&\equiv [s_{j+1}, ss_1^i] \\
&\equiv [s_{j+1}, s_1^i][s_{j+1}, s][s_{j+1}, s, s_1^i] \\
&\equiv [s_{j+1}, s_1^i]s_{j+2}.
\end{aligned}$$

Según nuestra hipótesis, G no es excepcional, por tanto, para $j+1 \leq m-2$ se tiene:

$$[s_{j+1}, s_1^i] \in [Y_{j+1}, Y_1] \leq Y_{j+3}.$$

Esto demuestra nuestra afirmación.

Como $Y_{m-1} = 1$, se tiene lo siguiente:

$$s_{m-1} = [s_2, \overbrace{ss_1^i, \ldots, ss_1^i}^{(m-3)}] \in Y_{m-2}(H).$$

Según hemos visto en 2.1.14, G/Y_{m-1} no es excepcional, y, por 2.1.7 se verifica que $Y_{m-2} = \langle Y_{m-1}, s_{m-2} \rangle$. Como $s \notin Y_1 = C_G(Y_{m-2})$, se tiene que $[s_{m-2}, s] = s_{m-1} \neq 1$. De aquí se obtiene que H tiene al menos clase $m-2$. Como H tiene orden p^{m-1}, se sigue que H tiene trivialmente clase $m-2$. □

Como hemos visto en las demostraciones anteriores, el sistema formado por los elementos $s \in G \setminus \big(Y_1 \cup C_G(Y_{m-2})\big)$, $s_1 \in Y_1 \setminus Y_2$ y $s_i = [s_{i-1}, s] \in Y_i \setminus Y_{i+1}$ para $i \in \{2, \ldots, m-1\}$ tiene una importancia capital. Por ello presentamos la siguiente definición.

Definición 2.1.30. Denominamos *G-sistema generador* al vector

$$(s, s_1, \ldots, s_{m-1}).$$

2.2. Cotas anteriores

Blackburn probó las siguientes cotas para $c(G)$:

Teorema 2.2.1 ([2, Theorem 2.11])**.** *Si m es un número impar, entonces $c(G) \geq 1$.*

Teorema 2.2.2 ([2, Theorem 3.8])**.** *Si $m \geq p+2$, entonces $c(G) \geq 1$.*

Teorema 2.2.3 ([2, Theorem 3.12])**.**

1. *Si G es metaabeliano, entonces $c(G) \geq m-p-1$.*
2. *Si $\mathcal{G}_a$ denota la familia de los p-grupos de clase maximal cuyo mayor subgrupo normal abeliano es Y_a, entonces $G \in \mathcal{G}_a$ y $a \geq 3$ implican que $c(G) \geq m-p-2a+4$.*

Teorema 2.2.4 ([2, Theorem 3.13]). *Si $p = 3$ y $m \geq 4$, entonces $c(G) \geq m - 4$.*

Teorema 2.2.5 ([2, Theorem 3.14]). *Si $p = 5$ y $m \geq 6$, entonces $c(G) \geq [(m-5)/2]$, donde $[x]$ denota la parte entera del número real x, esto es, el mayor entero menor o igual que x.*

La mayor parte de la notación que vamos a usar en lo sucesivo proviene de la tesis doctoral de **Raymond Shepherd** ([22]). Algunas de las cotas establecidas por R. Shepherd para $c(G)$ fueron las siguientes:

Teorema 2.2.6 (Shepherd). *Si $m \geq 4$, entonces $c(G) \geq [(m - 3p + 7)/2]$ y, por consiguiente, $Y_1(G)$ tiene clase a lo sumo 3.*

En [31], A. Vera López y G. A. Fernández Alcober obtienen cotas inferiores para el grado de conmutatividad de un p-grupo de clase maximal de orden p^m.

En todas las cotas conocidas con anterioridad, aparece el primo p, y resultan casi inútiles para valores pequeños de m. Se introduce un nuevo invariante b relacionado con la estructura de conmutadores de G y se obtiene una cota que depende sólo de b y de m, no de p. Como consecuencia, se acota la longitud derivada de G y la clase de nilpotencia de un cierto subgrupo maximal en términos de b. Por otro lado, se generalizan algunos resultados de **Blackburn**. Se dan ejemplos que muestran la exactitud de las cotas.

Como $[s_i, s_j] \in Y_{i+j+c}$, podemos definir $\alpha_{i,j} \in \mathrm{GF}(p)$ para $i + j \leq m - c - 1$ por la relación

$$[s_i, s_j] \equiv s_{i+j+c}^{\alpha_{ij}} \pmod{Y_{i+j+c+1}}.$$

Podemos suponer en lo sucesivo que $p \geq 3$, al ser los 2-grupos de clase maximal muy conocidos ([12, III-11.9]). Los α_{ij} satisfacen las siguientes condiciones:

(C1) Al menos un $\alpha_{i,j}$ es no nulo.

(C2) $\alpha_{i,j} = -\alpha_{j,i}$ y $\alpha_{i,i} = 0$ cuando está definido.

(C3) $\alpha_{i,j} = \alpha_{i+1,j} + \alpha_{i,j+1}$ para $i + j \leq m - c - 2$.

(C4) $\alpha_{i,j} = \alpha_{i+p-1,j} = \alpha_{i,j+p-1}$ para $i + j \leq m - c - p$ (periodicidad módulo $p - 1$).

(C5) Para $i+j+k \leq m-2c-1$, se tiene que

$$\alpha_{i,j}\alpha_{i+j+c,k} + \alpha_{j,k}\alpha_{j+k+c,i} + \alpha_{k,i}\alpha_{k+i+c,j} = 0$$

(identidad de **Jacobi**).

Notemos que **(C1)** y **(C4)** implican que no podemos tener $\alpha_{i,j} = 0$ para $p-1$ valores consecutivos de j.

Las relaciones de conmutadores de G derivadas a partir de los generadores s_i son más simples cuando $c(G)$ es grande. Por esta razón, es interesante obtener cotas inferiores generales para $c(G)$, que nos permitirían simplificar los cálculos cuando manejemos p-grupos de clase maximal.

Blackburn indicó que su cota $c(G) \geq m-p-1$ para $a=2$ (esto es, G metaabeliano) y $c(G) \geq m-p-2a+4$ para $a \geq 3$ es probablemente la mejor posible para $a < p$. Más tarde, Shepherd ([22]) y Leedham-Green y McKay ([14]) obtuvieron independientemente la cota $c(G) \geq [(m-3p+7)/2]$ que depende sólo de m y p. Esta cota se puede mejorar a $2c \geq m-2p+5$ cuando Y_1 tiene clase de nilpotencia 2. Esto fue probado por Leedham-Green y McKay ([16, Theorem 9.7]) como consecuencia de su construcción de todos los p-grupos de clase maximal con Y_1 de clase 2. Redujeron el problema al cálculo de $\mathrm{Hom}_{C_p}(\mathcal{O}/\mathcal{P}^{t-1} \wedge \mathcal{O}/\mathcal{P}^{t-1}, \mathcal{O}/\mathcal{P}^{m-t})$ para ciertos m y t, donde $\mathcal{O}$ es el anillo de los enteros en el p-ésimo cuerpo ciclotómico generado por una raíz primitiva compleja p-ésima de la unidad θ, $\mathcal{P}$ es el ideal generado por $\kappa = \theta - 1$ y C_p actúa mediante multiplicación por θ. Se da una prueba directa de este resultado en la que sólo aparecen las propiedades básicas de los $\alpha_{i,j}$ y la fórmula del resultado de Shepherd [22, Lemma 2.3]:

Teorema 2.2.7.

$$\alpha(i,j) = \sum_{\nu=i}^{[(i+j-1)/2]} (-1)^{\nu-i} \binom{j-\nu-1}{\nu-i} \alpha(\nu, \nu+1) \quad \textit{para } i+j \leq m-c-1, \tag{2.21}$$

Este resultado se deriva fácilmente de (C3).

Teorema 2.2.8 (Leedham-Green, McKay)**.** *Sea $p \geq 5$, y supongamos que Y_1 tiene clase de nilpotencia 2. Excepto en el caso en que $|G| = 5^6$ y $c(G) = 0$, tenemos que $2c(G) \geq m-2p+5$.*

Uno de nuestros objetivos en el estudio de los p-grupos de clase maximal es el cálculo del número de clases de conjugación y los órdenes de los diferentes centralizadores de sus elementos. Se ha intentado resolver este problema para grupos de orden pequeño ($|G| \leq p^9$), pero p arbitrario. Para este propósito, las cotas antes mencionadas dan escasa información sobre $c(G)$, ya que p aparece afectada por un signo menos en todas ellas. Este hecho nos ha llevado a buscar nuevas cotas inferiores para $c(G)$ que son independientes de p y bastante buenas para valores pequeños de m. En lo siguiente, exponemos los resultados publicados de nuestra búsqueda en esta dirección.

Si G es un p-grupo de clase maximal, se define

$$b = b(G) = \text{mín}\{k \mid [Y_i, Y_j] \leq Y_{i+j+c+1} \quad \text{para todo } i,\, j \geq k\}.$$

Es claro que $b = 1$ si, y sólo si, Y_1 es abeliano, esto es, $c(G) = m - 2$. Luego podemos restringir nuestra atención al caso $b \geq 2$. Obviamente, $b \leq a$ y, si $t = [(m-c)/2]$, de $[Y_t, Y_t] = 1$ derivamos que $b \leq t$, esto es, $c(G) \leq m - 2b$. En particular, se tiene siempre que $b \leq m/2$ y el valor de b está controlado por m.

Podemos precisar ahora algunos de los tipos de cotas que podemos buscar: cotas para $c(G)$ del tipo

$$c(G) \geq \frac{m - \lambda b - \mu}{\nu},$$

donde λ, $\nu \in \mathbb{N}$ y $\mu \in \mathbb{Z}$, que llamaremos cotas de tipo (m, b), en contraste con cotas como

$$c(G) \geq \frac{m - \lambda p - \mu}{\nu},$$

a las que nos referiremos como cotas de tipo (m, p).

A partir de la definición de b, tenemos que $\alpha_{i,j} = 0$ para $i,\, j \geq b$ cuando estén definidas. Seguidamente, veremos que el resto de las $\alpha_{i,j}$ pueden expresarse en términos de los $\alpha_{k,b}$ con $1 \leq k \leq b-1$. Los números combinatorios que utilizaremos son los llamados *números combinatorios generalizados*, definidos mediante la regla siguiente:

Definición 2.2.9. 1. Si $s \geq 1$,

$$\binom{r}{s} = \frac{r(r-1)\cdots(r-s-1)}{s!}.$$

2. Si $s = 0$,
$$\binom{r}{s} = 1.$$

3. Si $s < 0$,
$$\binom{r}{s} = 0.$$

Teorema 2.2.10. *Sea* $1 \leq i \leq b-1$.

1. *Si* $b \leq j \leq m-c-i-1$ *entonces*
$$\alpha(i,j) = \sum_{k=0}^{b-i-1} (-1)^k \binom{j-b}{k} \alpha(i+k,b). \tag{2.22}$$

2. *Si* $i < j \leq b-1$ *entonces*
$$\begin{aligned}\alpha(i,j) = &\sum_{k=0}^{j-i-1} \binom{b-j-1+k}{k} \alpha(i+k,b) \\ &+ \sum_{k=j-i}^{b-i-2} \left(\binom{b-j-1+k}{k} - \binom{b-j-1+k}{k+i-j} \right) \alpha(i+k,b). \end{aligned} \tag{2.23}$$

En [34], A. Vera López y B. Larrea introducen la siguiente clase, que desempeña un papel importante en su estudio:

Definición 2.2.11. Se define la clase $\mathcal{F}$ como sigue:

$$\mathcal{F} = \{G \mid G \text{ es un } p\text{-grupo de clase maximal y } c(G) \neq c\big(G/Z(G)\big)\}$$

En estas circunstancias, en [31, Lemma 4] se prueba el siguiente resultado:

Lema 2.2.12. *Si* $G \in \mathcal{F}$, *entonces* $b = a$ *y* $c(G) = m - 2b$.

El siguiente teorema corresponde a [31, Theorem 5]:

Teorema 2.2.13. *1. Sea* $1 \leq i \leq b-1$. *Si* $\alpha(i,j) \neq 0$ *para* $b \leq j \leq m-c-i-1$, *entonces* $c(G) \geq m-p-b-i+2$.

2. *Si* $b = 2$ *entonces* $c(G) \geq m - p - 1$. *Además, la relación* $c(G) \geq 1$ *se tiene siempre.*

3. *Si* $b \geq 3$, *entonces* $c(G) \geq m-p-2b+4$. *Por tanto,* $c(G) = m-p-2b+k$ *con* $4 \leq k \leq p$.

Nótese como las dos últimas partes de este teorema muestran que podemos substituir a y b en las cotas de Blackburn. Observamos que la desigualdad $c(G) \geq m - p - 2b + 3$ para $b \geq 2$ está probada también, argumentando de modo diferente, en [22, Lemma 1.21]. Las cotas para $b = 2$ no pueden ser mejoradas, ya que Miech en [18] probó que, para $p \geq 3$ y $m > p + 1$, existen p-grupos metaabelianos de clase maximal de orden p^m y con grado de conmutatividad $c(G) = m - p - 1$. Sin embargo, la cuestión de si la cota para $b \geq 3$ puede ser mejorada aún está abierta.

Corolario 2.2.14. *Supongamos que* $b \geq 2$. *Entonces* $a = b = 2$ *ó* $a \in [b, b + (p - 5)/2]$, *según sea* $p = 3$ *ó* $p \geq 5$.

Teorema 2.2.15. *Si* $b \geq 3$, *entonces* $c(G) \geq (m - 3b + 3)/2$.

La cota $c(G) \geq m-p-2b+4$ puede ser mejorada para ciertos valores de m.

Corolario 2.2.16. *Supongamos que* $b \geq 3$ *y sea* $4 \leq k \leq p$. *Si* $m \leq 2(p - k) + b + 4$, *entonces* $c \geq m - p - 2b + k$.

La cota $c(G) \geq (m - 3b + 3)/2$ puede ser mejorada para $b \geq 5$, como se ve en el siguiente teorema.

Teorema 2.2.17. *Si* $b \geq 5$, *entonces* $c(G) \geq (m - 3b + 4)/2$. *Más aún, a menos que* $b = 5$, $c = 1$ *y* $m = p = 13$, *tenemos que* $c(G) \geq (m - 3b + 5)/2$.

2.3. Construcción de álgebras de Lie

Teorema 2.3.1. *Sea* K *un cuerpo y* L *un álgebra de dimensión finita sobre* K, *con* $m = \dim L \geq 4$. *Sea* $(e_0, e_1, \ldots, e_{m-1})$ *una base de* L, *y definamos* $e_i = 0$ *ara* $i \geq m$. *Denotemos la multiplicación de* L *por* $[,]$, *y pongamos*

$$\mathcal{J}(i, j, k) = [e_i, e_j, e_k] + [e_j, e_k, e_i] + [e_k, e_i, e_j]$$

para i, j, $k \geq 0$ *(adoptamos el convenio* $[x, y, z] = \big[[x, y], z\big]$*). Supongamos que existen enteros* a *y* c, *con* $0 \leq c \leq m - 2$ *y* $1 \leq a \leq (m - c)/2$, *tales que* $[,]$ *satisface las siguientes condiciones:*

1. $[e_i, e_i] = 0$ *para* $i \geq 0$.
2. $[e_i, e_j] = -[e_j, e_i]$ *para cada* $0 \leq i < j$.
3. $[e_i, e_0] = e_{i+1}$ *para cada* $i \geq 1$.
4. $[e_i, e_j] \in \langle e_{i+j+c} \rangle$ *para* $1 \leq i \leq a-1$, $i < j$.
5. $[e_i, e_j] = 0$ *para* i, $j \geq a$.
6. $\mathcal{J}(0, i, j) = 0$ *para* $1 \leq i < j$.
7. $\mathcal{J}(i, j, k) = 0$ *para* $1 \leq i < j \leq a-1$, $j < k$, $i + j + k = m - 2c - 1$.

Entonces L es un álgebra de Lie nilpotente de clase maximal.

Este teorema nos dice que, bajo las condiciones especiales 3, 4 y 5 para los productos de Lie básicos, no necesitamos comprobar la identidad de Jacobi para todo el rango de valores $1 \leq i < j \leq a-1$, $j < k$, $i+j+k = m-2c-1$.

2.4. Cotas en función de b y l

En [30], A. Vera-López, J. M. Arregi y F. J. Vera-López obtienen nuevas cotas inferiores para el grado de conmutatividad de un p-grupo de clase maximal. Estas cotas muestran la relación entre dos nuevos invariantes $\big(b(G), v(G)\big)$ asociados a la estructura normal de G (que se usan en el cálculo de las relaciones definitorias de G) y el grado de conmutatividad.

Se define $v = v(G) = \text{mín}\{k \in [2, m-c-2] \mid [Y_1, Y_k]\}$. Es claro que $b = 1$ si, y sólo si, Y_1 es abeliano, esto es, $c(G) = m - 2$. Luego podemos restringir nuestra atención al caso $b \geq 2$.

En este artículo se prueba que $v = v(G)$ es un número par. Denotaremos $v = 2l$. Entonces se tienen las relaciones $l \leq b - 1$ y $2l \leq p - 1$. Se define, para cada entero no negativo n,

$$\binom{x}{n}^{*} = \begin{cases} 1 & \text{si } n = 0, \\ x(x-1)\cdots(x-n+1) & \text{si } n \geq 1. \end{cases}$$

Para calcular las relaciones definitorias de G, se fijan a priori las estructuras de conmutadores de G, esto es, la función $f = f(i, j, c)$ tal que $[Y_i, Y_j] =$

$Y_{f(i,j,c)}$, y, en consecuencia, los invariantes b y l están prefijados. Por tanto, es importante conocer a priori tanta información como sea posible sobre c en relación a m, que nos permitiría simplificar cálculos, esto es, queremos encontrar funciones $h_i = h_i(b,l)$, $1 \leq i \leq 3$, que son independientes de m y que satisfacen las desigualdades

$$p \geq h_1(b,l), \quad c \geq h_2(b,l) \Longrightarrow c \geq \big(m - h_3(b,l)\big)/2, \tag{2.24}$$

y lo más pequeñas posibles entre las que verifiquen estas condiciones. Se prueba que

$$h_1(b,l) = \text{máx}\big(2(b-l), b+l\big), \quad h_2(b,l) = 2(b-l)-3, \quad h_3(b,l) = 3b-(l+1)$$

satisfacen (2.24) en el caso $b \leq 2l-1$. Además, para $l = b-1$ y $b \geq 3$, las hipótesis son válidas, y existen ejemplos que satisfacen la igualdad $c = (m-3b+l+1)/2$ para cada p y c, por tanto,

$$c \geq (m-3b+l+1)/2$$

es la mejor cota posible. Por otro lado, la diferencia $b-l$ es pequeña para los grupos que tienen exponente pequeño. Es interesante obtener información sobre estos grupos. En este sentido, para $b-l = i$, $1 \leq i \leq 3$, y para cualquier p y cualquier c, se prueba que la desigualdad $c \geq (m-3b+l+1)/2$ es también válida. Finalmente, usando que $c \geq (m-3b+5)/2$, se prueba que $c \geq (m-3b+l+1)/2$ es válida, incluso en el caso $b-l = 4$.

Recordemos que si $G \in \mathcal{F}$, entonces se tiene la siguiente desigualdad:

$$c \geq m - p - 1.$$

Claramente, $\mathcal{F} \subseteq \mathcal{T}_1$, donde

$$\mathcal{T}_i = \{G \mid l(G) = b(G) - i\},$$

para $i \in \{1, \ldots, b-1\}$. Blackburn [2] mostró que

$$c(G) \geq m - p - 2a + 4,$$

siendo $a \geq 3$. Obviamente, $b \leq a$ y $c \leq m - 2b$. En [31], las desigualdades $c \geq m-p-1$ y $c \geq m-p-2b+4$ se prueban para los casos $b = 2$ y $b \geq 3$, respectivamente. Si $b = b(G) = 2$, es claro también que $G \in \mathcal{T}_1$. Más aún, es

obvio que si $b = b(G) \leq 5$, entonces $G \in \bigcup_{i=1}^{4} \mathcal{T}_i$. En este artículo se obtiene la siguiente desigualdad:

$$c \geq m - p - 2b + 2l + 1$$

si $p \geq \text{máx}\big(2(b-l)-1, b+l\big)$ y $b \leq 2l$. En particular, esta desigualdad se da en el caso $p \geq 2b - 1$ y $b \leq 2l$. Por otra parte, si

$$H_0 = G > H_1 > \cdots > H_{b-2}$$

es una cadena de p-subgrupos de clase maximal de G tal que $|H_i| = p^{m-i}$, entonces se dan las siguientes desigualdades:

$$c \geq m - p - 2b + 2l(H_i) + 1$$

para cada $i \in [0, b-2]$ tal que $b \leq 2l(H_i) + i$ y $p \geq 2(b-i) - 1$ (observemos que $l(H_i) \geq 2$ si, y sólo si, $\alpha_{i+1,i+2} = 0$). Por otro lado, en el caso $p + 3 \leq 2b \leq 2l + (p+1)/2$ la siguiente desigualdad es válida:

$$c \geq m - p - 2b + 2l(H_{b-((p+1)/2)}) + 1.$$

En el teorema 2.4 se prueba la siguiente desigualdad:

$$c \geq m - p - 2b + 2(b - l) + 1$$

si $b \leq 2l$. Estas desigualdades son especialmente interesantes cuando p es pequeño en relación a m, puesto que, en este caso, estas desigualdades dan más información que las desigualdades de tipo

$$c \geq m - h_3(b, l)2.$$

Finalmente, para $b - l$ pequeño, se obtienen las siguientes desigualdades: si $G \in \mathcal{T}_1$,

$$c \geq m - p - 1;$$

si $G \in \mathcal{T}_2$,

$$c \geq m - p - 3;$$

si $G \in \mathcal{T}_3$, entonces

$$c \geq \begin{cases} m - p - 4, & \text{si } l = 1; \\ m - p - 6, & \text{si } l = 2; \\ m - p - 5 & \text{si } l \geq 3. \end{cases}$$

Finalmente, si $G \in \mathcal{T}_4$, entonces tenemos

$$c \geq \begin{cases} m-p-6, & \text{si } l=1; \\ m-p-8, & \text{si } l=2; \\ m-p-10, & \text{si } l=3; \\ m-p-7, & \text{si } l \geq 4 \text{ y } 2l \neq p-3; \\ m-p-12, & \text{si } l=4 \text{ y } 2l=p-3; \\ m-p-9, & \text{si } l \geq 5 \text{ y } 2l=p-3. \end{cases}$$

Sea $\mathcal{TR}_G$ el siguiente triángulo matricial:

$$\mathcal{TR}_G = \begin{array}{ccccccccc} \alpha_{1,m-c-2} & \alpha_{2,m-c-3} & & \cdots & & & \alpha_{m-c-3,2} & \alpha_{m-c-2,1} \\ \alpha_{1,m-c-3} & \alpha_{2,m-c-4} & & \cdots & & & \alpha_{m-c-3,1} & \\ \vdots & \vdots & & & \iddots & \iddots & & \\ \alpha_{1,5} & \alpha_{2,4} & \alpha_{3,3} & \alpha_{4,2} & \alpha_{5,1} & & & \\ \alpha_{1,4} & \alpha_{2,3} & \alpha_{3,2} & \alpha_{4,1} & & & & \\ \alpha_{1,3} & \alpha_{2,2} & \alpha_{3,1} & & & & & \\ \alpha_{1,2} & \alpha_{2,1} & & & & & & \\ \alpha_{1,1} & & & & & & & \end{array}$$

esto es, $\mathcal{TR}_G = \{\alpha_{i,j} \mid i,j \geq 1, i+j \leq m-c-1\}$. De las definiciones de $c(G)$ se sigue que $\mathcal{TR}_G \neq (0)$.

Para cada número natural $g \leq m-c-1$, definimos el siguiente subtriángulo de $\mathcal{TR}_G$:

$$\mathcal{TR}_g = \mathcal{TR}_g(G) = \{\alpha_{i,j} \mid i,j \geq 1, i+j \leq g\}.$$

El conjunto

$$\mathcal{TRF}_g = \{a_{i,j} \mid i+j=g\}$$

se llama g-ésima fila de la tabla $\mathcal{TR}_G$, y el conjunto

$$\mathcal{TRC}_g = \{\alpha_{g,j} \mid 1 \leq j \leq m-c-1-g\}, \quad g \leq m-c-2$$

recibe el nombre de g-ésima columna de $\mathcal{TR}_G$.

Lema 2.4.1. *Supongamos que $b \geq 2$. El invariante*

$$v(G) = \min\{j \in \{2,3,\ldots,m-c-2\} \mid \alpha_{1,j} \neq 0\}$$

satisface las siguientes condiciones:

1. $v = 2l$ *para algún natural* $l = l(G)$.

2. $l \leq b - 1$.

3. $2l \leq p - 1$. *Más aún, si* $2l + 2 \leq m - c - 1$, *entonces* $2l \leq p - 3$.

Lema 2.4.2. *1. La totalidad de valores de la* j*-ésima columna de la matriz* $\mathcal{TR}_G$ *sólo depende de los valores de la* $(j+1)$*-ésima columna y de cualquier valor prefijado en la* j*-ésima columna.*

2. La totalidad de valores de la t*-ésima fila de la matriz* $\mathcal{TR}_G$ *sólo depende de los valores de la* $(t-1)$*-ésima fila y de cualquier valor prefijado de la* t*-ésima fila.*

Corolario 2.4.3. *1. Si la* j*-ésima columna de la matriz* $\mathcal{TR}_G$ *tiene un cero, entonces todos los valores de esta columna sólo dependen de los valores de la* $(j+1)$*-ésima columna de* $\mathcal{TR}_G$.

2. Si la t*-ésima columna de la matriz* $\mathcal{TR}_G$ *tiene un cero, entonces todos los valores de esta fila sólo dependen de los valores de la* $(t-1)$*-ésima fila de* $\mathcal{TR}_G$.

Corolario 2.4.4. *1. Si* $\{\alpha_{i_1,j_1}, \ldots, \alpha_{i_k,j_k}\}$ *es un conjunto de representantes de la* u_w*-ésima fila,* $w = 1, \ldots, k$, *esto es, si* $i_w + j_w = u_w$, $w = 1, \ldots, k$, *entonces todos los valores de* $\mathcal{TR}_G$ *pueden darse en términos de los valores de este conjunto, esto es, son variables que generan todos los valores de la tabla* $\mathcal{TR}_G$. *Además,* $k \leq b - 1$.

2. Si $\{\alpha_{v_1,z_1}, \ldots, \alpha_{v_s,z_s}\}$ *es un conjunto de representantes de la* v_w*-ésima columna,* $w = 1, \ldots, s$, *entonces todos los valores de* $\mathcal{TR}_G$ *pueden darse en términos de los valores de este conjunto, esto es, son variables que generan todos los valores de la tabla* $\mathcal{TR}_G$.

Proposición 2.4.5. *Se tienen las siguientes afirmaciones:*

1. *Si* $v_s < m - c - 2$, *entonces* $c \geq m - p - v_s$.

2. *Si* $v_1 \geq p$, *entonces* $c \geq m - p - v_1 + 1$.

3. *Si* $v_1 \leq p - 1$, *entonces* $c \geq m - 2(p-1)$.

Lema 2.4.6. *Para cada número natural n tal que $2l+n \leq m-c-1$, tenemos:*

$$\alpha_{i,2l+n-i} = \sum_{g=1}^{[(n+1)/2]} (-1)^{l-i-g+1} \binom{l-i+n-g}{n-(2g-1)} x_g \tag{2.25}$$

Lema 2.4.7. *Sean l, e, d enteros con $d \geq e-1 \geq 0$. Sea $A = (a_{i,j}) \in \mathrm{Mat}(e \times e, \mathbb{Z})$ la matriz definida por*

$$a_{i,j} = \begin{cases} \binom{l+d+i-j-1}{2i-j+d-1}^{*} & \text{si } j = 1, \\ \binom{l+d+i-j-1}{2i-j+d-1}^{*} \binom{2l-2i+2j-1}{2j-3}^{**} & \text{en otro caso.} \end{cases}$$

Entonces

$$\det A = (-1)^{e(e-1)/2} \prod_{k=2}^{e-1} k! \prod_{t=0}^{e-2} (4l+2d-1-2t)^{e-t-1} \prod_{i=1}^{e} \binom{l+d+i-e-1}{2i-e+d-1}^{*}.$$

Lema 2.4.8 (periodicidad módulo $c+b-1$)**.** *Los valores de la parte*

$$[\alpha_{1,b}, \ldots, \alpha_{1,m-c-2}]$$

de la primera columna de la tabla $\mathcal{TR}_G$ tienen periodicidad $c+b-1$, esto es,

$$\alpha_{1,u} = \alpha_{1,u-(c+b-1)} \quad \text{para todo } u \in [2b-1+c, m-c-2].$$

Obviamente, de la relación

$$\alpha_{k,b+e} = \alpha_{k,2b-1+c+e} \qquad \text{para todo } k \in [1, m-2c-2b-e],$$

que aparece en la demostración de este lema, se sigue que la periodicidad $c+b-1$ es verdadera para todos los valores de las columnas del subtriángulo de $\mathcal{TR}_G$ que tienen como vértices $(1, m-c-2)$, $(m-c-b-1, b)$ y $(1, b)$. De la definición de b se tiene que $\alpha_{i,j} = 0$ para i, $j \geq b$ cuando esta expresión tiene sentido. En [31] se vio que el resto de los $\alpha_{i,j}$ pueden ser expresados en términos de los $\alpha_{k,b}$ con $1 \leq k \leq b-1$, esto es, para $1 \leq i \leq b-1$ se tienen las siguientes igualdades:

$$\alpha_{i,j} = \sum_{g=0}^{b-i-1} (-1)^g \binom{j-b}{g} \alpha_{i+g,b}, \tag{2.26}$$

para cada $j \in [b, m-c-i-1]$, y

$$\begin{aligned}\alpha_{i,j} = &\sum_{g=0}^{j-i-1} (-1)^g \binom{b-j-1+g}{b-j-1} \alpha_{i+g,b} \\ &+ \sum_{g=j-i}^{b-i-2} \left(\binom{b-j-1+g}{g} - \binom{b-j-1+g}{g+i-j} \right) \alpha_{i+g,b},\end{aligned} \tag{2.27}$$

para $1 \leq j < b$.

Lema 2.4.9. *Supongamos que $c = (m-2b-1)/2$. Sea $y_j = \alpha_{j,b}$. Entonces se verifican las siguientes igualdades:*

$$-y_j \left(\sum_{g=0}^{b-i-1} (-1)^g j + cgy_{i+g} \right) + y_i \left(\sum_{g=0}^{b-j-1} (-1)^g c + igy_{j+g} \right) = 0, \tag{2.28}$$

para i, j cualesquiera que satisfagan $1 \leq i < j \leq b-1$, $b-c \leq i+j \leq b$.

Lema 2.4.10. *Supongamos que $c = m-p-2b+4$, $p \geq 7$ y $b \geq 6$. Sea $x = \alpha_{b-1,b}$. Entonces tenemos que*

$$\alpha_{b-2,b} = -3x, \alpha_{b-3,b} = 2x, \alpha_{b-4,b} = \alpha_{b-5,b} = 0.$$

Lema 2.4.11. *Supongamos que $b = 6$ y $c = (m-2b-1)/2 \geq 3$. Entonces $\prod_{i=0}^{b-1}(c+i) \neq 0$, y $\alpha_{1,b} = \alpha_{1,6} \neq 0$.*

Lema 2.4.12. *Supongamos que $b = 6$. Entonces $c \neq (m-2b-1)/2$.*

Teorema 2.4.13. *Supongamos que $b \geq 6$. Entonces*

$$c \geq (m-3b+6)/2.$$

Lema 2.4.14. *Sean l, c, e números naturales tales que $c \geq 2e-3$. Sea $A = (a_{ij})$, con*

$$a_{ij} = \binom{l+c-(i-1)}{c-2i+j+2}$$

para $1 \leq i, j \leq e$. Entonces

$$\det(A) = \frac{\prod_{k=1}^{e-1} k! \cdot \prod_{k=1}^{e-1} \prod_{j=1}^{e-k} (2l+c-j+k) \cdot \prod_{k=1}^{c+3-2i} (l+c+2-i-k)}{\prod_{i=1}^{e} (c-2i+e+2)!}.$$

Teorema 2.4.15. *Supongamos que* $p \geq \max\big(2(b-l)-1, b+l\big)$ *y* $b \leq 2l$. *Entonces se tiene la siguiente desigualdad:*

$$c \geq m - p - 2b + 2l + 1.$$

Corolario 2.4.16. *Sea* $H_0 = G > H_1 > \cdots > H_{b-2}$ *una cadena de p-subgrupos de clase maximal de* G *con* $|H_i| = p^{m-i}$, $i \in \{0, 1, \ldots, b-2\}$. *Se tienen entonces las siguientes desigualdades:*

$$c \geq m - p - 2b + 2l(H_i) + 1$$

para cada $i \in [0, b-2]$ *tal que* $b \leq 2l(H_i) + i$ *y* $p \geq 2(b-i) - 1$.

Corolario 2.4.17. *Si* $p + 3 \leq 2b \leq (p+1)/2 + 2l$, *se tiene la siguiente desigualdad:*

$$c \geq m - p - 2b + 2l(H_{b-\frac{p+1}{2}}) + 1.$$

Teorema 2.4.18. *Si* $b \leq 2l$, *entonces*

$$c \geq m - p - 2b + 2(b-l) + 1.$$

Teorema 2.4.19. *Supongamos que se dan las siguientes condiciones:*

$$p \geq \max\big(2(b-l)-1, b+l\big), \quad c \geq 2(b-l) - 3 \quad y\ b \leq 2l - 1.$$

Entonces tenemos que

$$c \geq \frac{m - 3b + l + 1}{2}.$$

A continuación se estudia la familia $\mathcal{T}_1$.

Teorema 2.4.20. *Supongamos que* $l = b - 1$. *Se tiene entonces la siguiente desigualdad:*

$$c \geq m - p - 1.$$

Corolario 2.4.21. *Si* $G \in \mathcal{F}$, *entonces se tiene la desigualdad* $c \geq m - p - 1$.

Teorema 2.4.22. *Supongamos que* $l = b - 1$, *y sea* $x_1 = \alpha_{l,l+1}$. *Entonces los valores de la tabla* $\mathcal{TR}_G$ *son los siguientes:*

$$\alpha_{i,2l+n-i} = \begin{cases} (-1)^{l-i}\binom{l+n-i-1}{n-1}x_1, & \text{para } n \geq 1,\ 2l + n \leq m - c - 1; \\ 0, & \text{para } n \in [-2l+2, 0]. \end{cases}$$

Teorema 2.4.23. *Supongamos que $b \geq 3$ y $l = b - 1$. Entonces $c \geq (m - 2b)/2$.*

Teorema 2.4.24. *Sea K un cuerpo tal que $\operatorname{car} K \neq 2$ y sea L un K-espacio vectorial de dimensión m. Sea $\{e_0, e_1, \ldots, e_{m-1}\}$ una K-base de L. Sean v, c, l números enteros tales que*

$$v \geq 0, \quad 0 \leq c \leq m - 2, \quad l \leq (p-3)/2, \quad m = 2c + 2l + 2 - v.$$

Definimos $e_i = 0$ para cada $i \geq m$, y

1. $[e_i, e_0] = e_{i+1}$ *para* $i \geq 1$,
2. $[e_i, e_0] = -[e_0, e_i]$ *para* $i \geq 1$,
3. $[e_i, e_j] = (-1)^{i+1}\binom{l+w-i-1}{w-1} e_{i+j+c}$ *si* $i + j = 2l + w$, i, $j \geq 1$.

Entonces se tienen las siguientes afirmaciones:

1. $[e_i, e_i] = 0$ *para todo* $i \geq 0$.
2. $[e_i, e_j] = -[e_j, e_i]$, *para* $0 \leq i < j$.
3. $\rho(i, j, k) = [e_i, e_j, e_k] + [e_k, e_i, e_j] + [e_j, e_k, e_i] = 0$ *para* $0 \leq i < j < k$.

Entonces $(L, [,])$ es una K-álgebra de Lie de clase de nilpotencia maximal.

Corolario 2.4.25. *Sea $m \geq 6$ un número par menor o igual que p. Entonces existen p-grupos de clase maximal de orden p^m que satisfacen las condiciones*

$$c = (m - 2b)/2, \quad l = b - 1.$$

En el siguiente parágrafo, se estudia la familia $\mathcal{T}_2$.

Teorema 2.4.26. *Si $l = b - 2$, entonces $c \geq m - p - 3$.*

Teorema 2.4.27. *Supongamos que $l = b - 2$. Entonces para $2 \leq 2l + n \leq m - c - 1$, tenemos que*

$$\alpha_{i,2l+n-i} = (-1)^{l-i}\binom{l-i+n-1}{n-1} x_1 + (-1)^{l-i-1}\binom{l-i+n-2}{n-3} x_2,$$

donde $x_1 = \alpha_{l,l+1}$, $x_2 = \alpha_{l+1,l+2}$.

Teorema 2.4.28. *Supongamos que $l = b - 2$. Entonces $c \geq (m - 2b - 1)/2$.*

En el parágrafo 5 se estudia la familia $\mathcal{T}_3$.

Teorema 2.4.29. *Si $l = b - 3$, entonces tenemos que*

$$c \geq \begin{cases} m - p - 4 & si\ l = 1; \\ m - p - 6 & si\ l = 2; \\ m - p - 5 & si\ l \geq 3. \end{cases}$$

Teorema 2.4.30. *Supongamos que $l = b - 3$. Entonces, para $2 \leq 2l + n \leq m - c - 1$ tenemos que*

$$\alpha_{i,2l+n-i} = (-1)^{l-i}\left(\binom{l-i+n-1}{n-1}x_1 - \binom{l-i+n-2}{n-3}x_2 + \binom{l-i+n-3}{n-5}x_3\right),$$

donde $x_1 = \alpha_{l,l+1}$, $x_2 = \alpha_{l+1,l+2}$, $x_3 = \alpha_{l+2,l+3}$.

Teorema 2.4.31. *Supongamos que $l = b - 3$. Entonces*

$$c \geq (m - 2b - 2)/2.$$

Por último, se estudia la familia $\mathcal{T}_4$.

Teorema 2.4.32. *Supongamos que $l = b - 4$. Entonces tenemos:*

$$c \geq \begin{cases} m - p - 6 & si\ l = 1; \\ m - p - 8 & si\ l = 2; \\ m - p - 10 & si\ l = 3; \\ m - p - 7 & si\ l \geq 4\ y\ 2l \neq p - 3; \\ m - p - 12 & si\ l = 4\ y\ 2l = p - 3; \\ m - p - 9 & si\ l \geq 5\ y\ 2l = p - 3. \end{cases}$$

Teorema 2.4.33. *Supongamos que $l = b-4$. Entonces los valores de la tabla $\mathcal{TR}_G$ son los siguientes:*

$$\alpha_{i,2l+n-i} = \begin{cases} \sum_{g=1}^{4}(-1)^{l-i-g+1}\binom{l+n-i-g}{n+1-2g}x_g, & para\ n \geq 1, \\ & 2l + n \leq m - c - 1; \\ 0, & para\ n \in [-2l + 2, 0]. \end{cases}$$

Teorema 2.4.34. *Supongamos que $l = b - 4$. Entonces $c \geq (m - 2b - 3)/2$.*

2.5. Cotas en función de c_0 y l

En el artículo [28], A. Vera-López, J. M. Arregi y F. J. Vera-López obtienen nuevas cotas inferiores para el grado de conmutatividad de un p-grupo de clase maximal. Estas cotas muestran la relación entre el invariante $l(G)$ asociado a la estructura normal de G (que se usa en el cálculo de las relaciones definitorias de G) y el grado de conmutatividad.

Lema 2.5.1. *Sean e y t números naturales tales que $t \geq e$, y sea r una indeterminada. Consideremos la matriz $A = (a_{i,j})_{1\leq i,j\leq e}$, con*

$$a_{i,j} = \binom{r-i}{t-2i+j}.$$

Entonces

$$\det A = \frac{\prod_{i=1}^{\min(e,[(t+1)/2])}[r-i,t-2i+1]\cdot\prod_{k=2}^{e-1}k!\cdot\prod_{j=1}^{s-1}(2r-t+e-s+1-j)}{\prod_{i=1}^{e}(t-2i+e)!\cdot\prod_{s=2}^{e}\prod_{(t+e-s+2)/2\leq i\leq e}(r+e-s+1-i)}.$$

Nota 2.5.2. Tras una reordenación adecuada, la expresión para $\det A$ puede ser escrita de un modo más compacto como

$$\det A = \frac{F(r,t,e)}{F(t,t,e)}, \tag{2.29}$$

donde

$$F(r,t,e) = \prod_{1\leq w\leq t-1}(r-w)^{\min(w,t-e,t-w)} \times \prod_{t-e+1\leq w\leq t+e-3}(2r-w-1)^{\min([\frac{e-t+w+1}{2}],[\frac{e+t-w-1}{2}])}. \tag{2.30}$$

Se denota por c_0 la clase residual de $c(G)$ módulo $p-1$.

Lema 2.5.3. *Supongamos que $l \geq c_0+2$ y $2l+c_0+2 \leq m-2c-1$. Entonces*

$$2c \geq m-2l-c_0-2.$$

Demostración. En primer lugar, notemos que, bajo estas hipótesis,

$$\alpha_{1,2l+c_0+1} = \cdots = \alpha_{c_0+1,2l+c_0+1} = 0. \tag{2.31}$$

Tomemos un i tal que $l \leq i \leq c_0 + 1$. Las desigualdades $i < l < l + 1$ y la segunda hipótesis implican que $i + l + (l + 1) \leq 2l + c_0 + 2 \leq m - 2c - 1$. Por tanto, podemos aplicar la identidad de Jacobi:

$$\alpha_{i,l}\alpha_{i+l+c,l+1} + \alpha_{l,l+1}\alpha_{2l+i+c,i} + \alpha_{l+1,i}\alpha_{i+l+1+c,l} = 0.$$

Pero $i + l < i + (l + 1) < 2l + 1$, luego $\alpha_{i,l} = \alpha_{i,l+1} = 0$. Por definición de l, $\alpha_{l,l+1} \neq 0$. Por tanto, $\alpha_{2l+1+c,i} = 0$, luego $\alpha_{i,2l+1+c_0} = \alpha_{i,2l+1+c} = -\alpha_{2l+1+c,i} = 0$.

Los parámetros $\alpha_{i,j}$ pueden expresarse en función de algunos de ellos (véase [30, Lemma (1.6)]):

$$\alpha_{i,2l+n-i} = \sum_{j=1}^{[(n+1)/2]} (-1)^{l-i-j+1} \binom{l-i+n-j}{n-2j+1} x_j, \tag{2.32}$$

donde $x_i = \alpha_{l+i-1,l+i}$. Teniendo en cuenta que $\alpha_{1,j} = \alpha_{1,j_0}$, con $j \equiv j_0$ (mód $p-1$), $1 \leq j_0 \leq p-1$ y [30, Lemma (1.2)], se sigue que los valores de la tabla $\mathcal{TR}_G$ pueden darse en términos de las variables

$$x_l, x_{l+1}, \ldots, x_{(p-3)/2}$$

si $p \neq m - c - 1$, y en función de las variables

$$x_l, x_{l+1}, \ldots, x_{(p-3)/2}, x_{(p-1)/2}$$

en otro caso.

Tomando $n = c_0 + i + 1$, las relaciones (2.31) se transforman en

$$\sum_{j=1}^{[(c_0+i+2)/2]} (-1)^{l-i-j+1} \binom{l+c_0+1-j}{c_0+2+i-2j} x_i = 0, \qquad i \in \{1, \ldots, c_0+1\}.$$

Si $i = c_0 + 1$, el valor máximo del índice j es $[(2c_0 + 3)/2] = c_0 + 1$. De este modo, (2.32) representa un sistema de $c_0 + 1$ ecuaciones homogéneas en las

c_0+1 incógnitas x_j, $1 \leq j \leq c_0+1$. Como $x_l = \alpha_{l,l+1} \neq 0$, deducimos que el determinante de la matriz de coeficientes

$$d_{i,j} = (-1)^{l-i-j+1} \binom{l+c_0+1-j}{c_0+2+i-2j}$$

debe ser cero módulo p. Transponiendo y cambiando adecuadamente los signos en las filas y en las columnas, tenemos la matriz de coeficientes

$$a_{i,j} = \binom{l+c_0+1-i}{c_0+2+j-2i},$$

cuyo determinante debe ser también cero módulo p. Pero tal determinante es la expresión (2.29) para los parámetros $r = l+c_0+1$, $t = c_0+2$ y $e = c_0+1$, ya que se satisface la condición $c_0+2 \geq e$. Esto es,

$$\det A = \frac{F(l+c_0+1, c_0+2, c_0+1)}{F(c_0+2, c_0+2, c_0+1)},$$

donde

$$F(l+c_0+1, c_0+2, c_0+1) = \prod_{u=l}^{l+c_0} u \cdot \prod_{w=2l+1}^{2(l+c_0-1} w^{\text{mín}\{[(2(l+c_0)+1-w)/2],[(w-2l+1)/2]\}}$$

y

$$F(c_0+2, c_0+2, c_0+1) = \prod_{u=1}^{c_0+1} u \cdot \prod_{w=3}^{2c_0+1} w^{\text{mín}\{[(2c_0+3-w)/2],[(w-1)/2]\}}.$$

Hemos supuesto que $p > 2c_0+1$, con lo cual el denominador $F(c_0+2, c_0+2, c_0+1)$ es invertible en $\mathbb{F}_p$. De este modo, el numerador $F(l+c_0+1, c_0+2, c_0+1)$ debe ser un múltiplo de p, esto es, alguno de sus factores debe ser un múltiplo de p y, en consecuencia,

$$p \leq \big(l+c_0, 2(l+c_0)-1\big) = 2l+2c_0-1. \quad \square$$

Teorema 2.5.4. *Si G es un p-grupo de clase maximal tal que $c_0+2 \leq l \leq (p-2c_0-1)/2$, entonces*

$$2c \geq m-2l-c_0-2.$$

Ejemplos 2.5.5. 1. La afirmación del teorema 2.5.4 es falsa con las hipótesis $c_0 + 2 \leq l$ y $l \geq (p - 2c_0 + 1)/2$. Consideremos los p-grupos de clase maximal dados en [15, parágrafo 6], que satisfacen las siguientes condiciones:

$$[Y_1, Y_1] = Y_n, |G| = p^m, 2c = m - 2p + 5,$$
$$n = x(p-1) + 1, m = 2x(p-1) + 1.$$

Para estos p-grupos, $c = x(p-1) - p + 3 = n - p + 2$, de donde

$$c \equiv n - (p-1) + 1 \equiv n + 1 \equiv 2 \pmod{p-1},$$

luego $c_0 = 2$. Tenemos que

$$[Y_1, Y_{i+1}] \leq Y_{2i+1+c}, \quad [Y_i, Y_{i+1}] \leq [Y_1, Y_1] = Y_n$$

y se tiene que $2i + 1 + c \leq n - 1$ si, y sólo si, $2i \leq p - 4$. Por tanto, $2i \leq p - 5$ implica que $[Y_i, Y_{i+1}] < Y_{2i+1+c}$, esto es,

$$x_1 = x_2 = \cdots = x_{(p-5)/2} = 0, x_{(p-3)/2} \neq 0,$$

ya que las variables x_i, $i \in \{1, 2, \ldots, (p-3)/2\}$, generan la totalidad de la tabla $\mathcal{TR}_G$. En otras palabras, estos grupos satisfacen

$$2l = p - 3 = p - 2c_0 + 1, \quad c_0 = 2.$$

Es obvio que $2c = m - 2p + 5 \ngeq m - 2l - c_0 - 2 = m - (p-3) - 2 - 2$ y, si $p \geq 11$, entonces se tiene la primera condición: $l = (p-3)/2 \geq c_0 + 2 = 4$.

2. La afirmación del Teorema 2.5.4 no se satisface para $l < c_0 + 2$. La Tabla 3 de [32] contiene ejemplos de p-grupos de clase maximal (núm. 26, 27, 30, 31) que satisfacen $c = 1, m = 8, l = 1$ y $2c = 2 \cdot 1 \ngeq 8 - 2 \cdot 1 - 1 - 2 = 3$. Por tanto, la relación $2c \geq m - 2l - c_0 - 2$ de Shepherd no se satisface para el caso $l = c_0$.

3. En la tabla 3 de [32] se dan ejemplos de p-grupos de clase maximal (núm. 32–39) que satisfacen $m = 8$, $c = c_0 = 0$, $l = b - 1 = 3$, $2c = 0 = m - 2l - c_0 - 2$. Por tanto, la desigualdad de Shepherd no puede mejorarse.

Nota 2.5.6. 1. Bajo las hipótesis de este teorema, la condición

$$2c \geq m - 2l - c_0 - 2 \geq m - 3l \geq m - (3/2)(p-1)$$

se satisface. Para esta familia de p-grupos de clase maximal, esta cota mejora la de Shepherd, C.R. Leedham-Green y S. McKay: $2c \geq m - 3p + 7$.

2. Si $2c_0 = p - 2l - 1$ y $l \geq c_0 + 2$, entonces

$$\begin{aligned} 2c &\geq m - p + c_0 \quad \text{si } c_0 < c, \\ c &\geq m - p - 1 \quad \text{si } c_0 = c. \end{aligned}$$

Notamos también que si $l(G) \geq k \geq 2$, entonces $c(H_i) = c+i$ y $l(H_i) = l(G)-i$ para cada $i \in \{1, 2, \ldots, k-1\}$, donde $G = H_0 > H_1 > H_2 > \cdots > H_{b-1}$ es una cadena de p-grupos de clase maximal con $|H_i| = p^{m-i}$. Utilizando este resultado, deducimos el siguiente:

Corolario 2.5.7. *Supongamos que $l(G) \geq p + 1 - c_0$. Entonces $2c(G) \geq m - 2l - (p - 1 - c_0) - 2 \geq m - 3l$.*

En [30] se analizan algunas propiedades de los p-grupos de clase maximal que satisfacen la condición

$$b \leq 2l - 1.$$

La demostración del Teorema 2.5.4 utiliza la periodicidad módulo $p-1$ de los $\alpha_{i,j}$. Ahora se obtiene un resultado análogo para la periodicidad $c+b-1$ en el trapecio correspondiente de vértices $(1, b)$, $(1, m-c-2)$, $(b-1, b)$, $\big(b-1, m-c-1-(b-1)\big)$, de acuerdo con [30, Lema 1.8]. Se necesita el siguiente resultado:

Lema 2.5.8. *Supongamos que $2d_0 < p+1$, $b \leq 2l-1$, $4l \leq m-2c$ y $2l - b \geq d_0$, donde $d = c + 2b - 2l - 1 \equiv d_0 \pmod{p-1}$ y $d_0 \in [0, p-2]$. Entonces*

$$2l + 2d_0 - 3 \geq p.$$

Teorema 2.5.9. *Si G es un p-grupo de clase maximal tal que*

$$\max\big((b+1)/2, (b+d_0)/2\big) \leq l \leq (p - 2d_0 + 2)/2,$$

entonces:

$$2c \geq m - 4l + 1.$$

Ejemplo 2.5.10. Consideremos la familia de p-grupos de clase maximal mencionada después del Teorema 2.5.4. Tenemos que

$$[Y_{(n+p-4)/2}, Y_{(n+p-4)/2+1}] \leq Y_{n+p-3+c} = Y_{2n-1} = Y_m = 1,$$

con lo que $b \geq (n+p-4)/2 = m-c-1$, esto es, $c \geq m-2b-1$. Pero m es impar, luego necesariamente $c = m-2b-1$. Las igualdades $c+2b-2l-1 = m-2l-2 = (2x-1)(p-1)+1$ nos conducen a que $d_0 = 1$. Por otro lado, $b = (n+p-4)/2 = (x+1)(p-1)/2-1 \geq 2(p-1)/2-1 = p-2 > 2l = p-3$ y $2l+2d_0-3 = p-3-1 \ngeq p$. Esto significa que la afirmación del Lema 2.5.8 falla para $b > 2l$, $2d_0 < p+1$ y $4l \leq m-2c$. Con respecto al Teorema 2.5.9, notamos que

$$l = (p-3)/2 < (x+1)(p-1)/2 = (b+1)/2 = \text{máx}\big((b+1)/2, (b+d_0)/2\big),$$

$$l \leq (p-2d_0+2)/2 = p/2$$

y

$$2c = m-2p+5 \ngeq m-4l+1 = m-2(p-3)+1 = m-2p+7.$$

Por tanto, la afirmación del teorema falla para $l < \text{máx}\big((b+1)/2, (b+d_0)/2\big)$ y $l \leq (p-2d_0+2)/2$.

2.6. Cotas para $c_0 \leq 5$

En [29], A. Vera-López, J. M. Arregi y F. J. Vera-López continúan la línea del anterior artículo, [28]. Se denota $x_\lambda = \alpha_{\lambda,\lambda+1}$. Todos los $\alpha_{i,j}$ pueden ser expresados en términos de los x_λ para $\lambda \leq (p+3)/2$ en el caso $p \neq m-c-1$, y para $\lambda \leq (p-1)/2$ en otro caso.

Se prueba que la condición $2k+p-c_0-1 \leq m-2c-1$ implica que $x_1 = x_2 = \cdots = x_{k-1} = 0$ cuando se dan las condiciones $\alpha_{j,p-c_0} = \alpha_{1,p-c_0}$ para todo $j \leq c_0$ y $k+c_0 \leq p$.

Usando este resultado, se concluye que si $\alpha_{j,p-c_0} = \alpha_{1,p-c_0}$ para todo $j \leq c_0$, entonces

$$2c \geq m - 2l - p + c_0 - 1.$$

Como corolario se obtiene que esta última relación se da en el caso $c_0+1 \leq 2l$. En [22], Shepherd da una demostración diferente de este caso. Además, en el caso $2c \geq m - 2l - p + c_0 - 2$ (resp., $2c \geq m - 2l - p + c_0 - 3$, $2c \geq m - 2l - p + c_0 - 5$).

Shepherd probó en [22, 1.20, 3.3, 3.11, 3.18, 3.24] que si $c_0 = i$, $0 \leq i \leq 4$, y $p \geq 11$, entonces

$$2c \geq h_i(m, p),$$

donde $h_0(m, p) = m - p - 1$, $h_1(m, p) = m - p + 1$, $h_2(m, p) = m - 2p + 5 = h_3(m, p)$ y $h_4(m, p) = m - 2p + 7$. Como aplicación de nuestro resultado, se obtiene el resultado de Shepherd en los casos $c_0 \leq 4$ y analizamos también el caso $c_0 = 5$ probando las siguientes desigualdades:

1. Si $2l \leq p - 11$ y $p > 19$, entonces $2c \geq m - p - 2$.

2. Si $p - 5 \geq 2l \geq p - 9$ y $p > 19$, entonces $2c \geq m - 2l - p + c_0 - 1 \geq m - 2p + c_0 + 4$.

3. Si $2l = p - 1$, entonces $c \geq m - p - 1$.

4. Si $2l = p - 3$ ó $p \leq 19$, entonces $2c \geq m - 2p + c_0 + 3$.

En particular, se obtiene que $2c \geq \text{mín}(m - p - 2, m - 2l - p + c_0 - 1)$ para $1 \leq c_0 \leq 5$. Esta información puede ser utilizada en el cálculo de las relaciones definitorias de G. En consecuencia, probamos que para $c_0 = 5$, la clase de nilpotencia del subgrupo maximal Y_1 es como máximo 3, salvo en los siguientes casos excepcionales:

1. $\text{Cl}(Y_1) = 4$, $c = p + 4$, $m \leq 4p$, o

2. $c = 5$, $m \leq 2p + 2$.

Sea $\mathcal{T}_G$ el siguiente triángulo matricial:

$$\mathcal{T}_G = \begin{array}{ccc} \alpha_{1,m-c-2} & \alpha_{2,m-c-3} & \cdots \\ \alpha_{1,m-c-3} & \alpha_{2,m-c-4} & \cdots \\ \vdots & \vdots & \\ \alpha_{1,5} & \alpha_{2,4} & \\ \alpha_{1,4} & \alpha_{2,3} & \\ \alpha_{1,3} & & \\ \alpha_{1,2} & & \end{array}$$

esto es,

$$\mathcal{T}_G = \{\alpha_{i,j} \mid i, j \geq 1, i < j, i + j \leq m - c - 1\}.$$

De la definición de $c(G)$, se sigue que $\mathcal{T}_G \neq (0)$. Definimos $x_\lambda = \alpha_{\lambda,\lambda+1}$ para $2\lambda + 1 \leq m - c - 1$.

La propiedad de Bernoulli (C3) toma una de las siguientes formas:

$$\begin{aligned} \alpha_{i,j} &= \alpha_{i,j+1} + \alpha_{i+1,j}, \quad \text{si } i + j + 1 \leq m - c - 1; \\ \alpha_{i,j} &= \alpha_{i,j-1} - \alpha_{i+1,j-1}, \quad \text{si } j > 1; \\ \alpha_{i,j} &= \alpha_{i-1,j} - \alpha_{i-1,j+1}, \quad \text{si } i > 1. \end{aligned}$$

Recurrencia sobre $w \geq 0$ para cada una de estas tres fórmula nos lleva a las siguientes expresiones más generales:

$$\alpha_{i,j} = \sum_{u=0}^{w} \binom{w}{u} \alpha_{i+u,j+w-u}, \quad \text{si } i + j + w \leq m - c - 1; \tag{2.33}$$

$$\alpha_{i,j} = \sum_{u=0}^{w} (-1)^u \binom{w}{u} \alpha_{i+u,j-w}, \quad \text{si } j > w; \tag{2.34}$$

$$\alpha_{i,j} = \sum_{u=0}^{w} (-1)^u \binom{w}{u} \alpha_{i-w,j+u}, \quad \text{si } i > w. \tag{2.35}$$

Para $r + s + t \leq m - c - 1$, denotemos

$$\mathcal{R}(r, s, t) = \{\alpha_{i,j} \mid i \geq r,\ j \geq s,\ i + j \leq r + s + t\},$$

el triángulo ($\subset \mathcal{T}_G$) de vértices $\alpha_{r,s}$, $\alpha_{r,s+t}$,$\alpha_{r+t,s}$, $t \geq 1$. Las fórmulas (2.33), (2.34) y (2.35) permiten determinar los $\alpha_{i,j}$ en $\mathcal{R}(r,s,t)$ cuando se conocen los valores correspondientes a un lado de $\mathcal{R}(r,s,t)$. En particular, si los valores en un lado de $\mathcal{R}(r,s,t)$ son todos cero, entonces los valores en todo el triángulo son cero.

Supongamos que $i+j+p-1 \leq m-c-1$. Denotamos $z_i^{(j)} = \alpha_{i,p-c_0-1+j}$ y $z_i = z_i^{(1)} = \alpha_{i,p-c_0}$.

Por la periodicidad módulo $p-1$ y por (2.34), obtenemos

$$\alpha_{ij} = \alpha_{i,j+p-1} = \sum_{u=0}^{w} (-1)^u \binom{w}{u} \alpha_{i+u,j+p-1-w} = \sum_{u=0}^{w} (-1)^u \binom{w}{u} z_{i+u}^{(j+c_0-w)}, \tag{2.36}$$

para cada $w \geq 0$ tal que $j+p-1 > w$ y $w \leq j+c_0$. En particular, para $w = j+c_0-1$ tenemos:

$$\alpha_{ij} = \alpha_{i,j+p-1} = \sum_{u=0}^{j+c_0-1} (-1)^u \binom{j+c_0-1}{u} z_{i+u}. \tag{2.37}$$

En [18], los α_{ij} son expresados en función de los valores fundamentales $x_k = \alpha_{k,k+1}$ como sigue:

$$\alpha_{i,j} = \sum_{k=i}^{[(i+j-1)/2]} (-1)^{k-i} \binom{j-k-1}{k-i} x_k. \tag{2.38}$$

Supongamos que $i+j+1+p-c_0-1 \leq m-2c-1$. Aplicando la identidad de Jacobi a la terna $(i,j,p-c_0-1)$ y teniendo en mente la periodicidad módulo $p-1$, tenemos la factorización $r_{ij} = \alpha_{ij}(\alpha_{i+j+c_0,p-c_0-1} - \alpha_{i,p-c_0-1} - \alpha_{j,p-c_0-1}) = 0$, esto es,

$$\alpha_{ij}(z_{i+j+c_0}^{(0)} - z_1^{(0)} - z_j^{(0)}) = 0.$$

Lema 2.6.1. *Supongamos que $i+j+1+p-c_0-1 \leq m-2c-1$. Se tienen las siguientes afirmaciones:*

1. *$\alpha_{i+1,j} \neq 0 \neq \alpha_{i,j+1}$ implica que $z_i = z_j$.*
2. *$\alpha_{i,j} \neq 0 \neq \alpha_{i+1,j}$ implica que $z_i = z_{i+j+c_0}$.*
3. *$\alpha_{i,j} \neq 0 \neq \alpha_{i,j+1}$ implica que $z_j = z_{i+j+c_0}$.*

4. *Si* $j \geq 2$, $0 \notin \{\alpha_{i,j}, \alpha_{i,j+1}, \alpha_{i+1,j-1}, \alpha_{i+2,j-1}\}$ *y* $\alpha_{i+1,j} = 0$, *entonces* $z_j = z_{i+j+c_0} = z_{i+1}$.

En la tabla $\mathcal{T}_G$, el caso 1 corresponde a dos valores consecutivos no nulos en la misma fila, lo que denotaremos por

$$\bullet \quad \bullet$$

El caso 2 corresponde a dos valores no nulos consecutivos en una misma diagonal, lo que denotamos por

$$\begin{array}{cc} & \bullet \\ \bullet & \end{array}$$

El caso 3 corresponde a dos valores no nulos consecutivos en la misma columna, lo que denotaremos por

$$\begin{array}{c} \bullet \\ \bullet \end{array}$$

El caso 4 corresponde a la configuración siguiente:

$$\begin{array}{cccc} & \bullet & 0 & \bullet \\ a_{ij}= & \bullet & \bullet & \end{array}$$

Con esta notación, cálculos usando la propiedad de Bernoulli muestran que todas las posibles configuraciones para $i + j \leq 8$ son las de las tablas I, II, III dadas al final del artículo.

En cualquier caso, la existencia de valores no nulos consecutivos fuerza una ligadura en los elementos de la $(p - c_0)$-ésima diagonal.

Sabemos a priori la existencia de filas de $\mathcal{T}_G$ con muchos valores no nulos. Particularmente, las filas $i + j = 2l + 1$, $2l + 2$ de $\mathcal{T}_G$ (esta última cuando existe) tienen todos sus valore no nulos. Por tanto, aplicando el Lema 2.6.1, obtenemos relaciones entre los z_j.

Lema 2.6.2. 1. *En la fila* $i + j = 2l + 1$ *de* $\mathcal{T}_G$, *todos los valores son no nulos.*

2. *Supongamos que* $2l+2 \leq m-c-1$. *En la fila* $i+j = 2l+2$ *de* $\mathcal{T}_G$, *todos los valores son no nulos. Además, si* $2l + 2 + p - c_0 - 1 \leq m - 2c - 1$ *y* $l \geq 2$, *entonces* $z_j = z_1$ *para cada* $j \in [1, 2l] \setminus \{l\}$.

3. *Supongamos que* $2l+3 \leq m-c-1$. *En la fila* $i+j=2l+3$ *de* $\mathcal{T}_G$, *existe a lo sumo un cero.*

Se define $y_v = \alpha_{1,2l+v}$ para $0 \leq v \leq m-c-2l-2$. Entonces

Lema 2.6.3. 1. $\alpha_{i,j} = 0$ *para* $i+j \leq 2l$.

2. *Si* $2l+1 \leq i+j \leq m-c-1$, *entonces*

$$\alpha_{ij} = \sum_{v=0}^{\tau} (-1)^{v-j} \binom{i-1}{\tau - v} y_v,$$

con $\tau = i+j-2l-1$.

Lema 2.6.4. *Para* $l \leq i \leq (m-c-3)/2$ *se tienen las siguientes desigualdades:*

$$y_{\tau_i} = \sum_{v=0}^{\tau_i - 1} (-1)^{\tau_i - v + 1} \frac{i-1}{\tau_i - v} y_v, \quad \text{con } \tau_i = 2(i-l)-1.$$

Lema 2.6.5. *Supongamos que* $2l+p-1 \leq m-c-1$ *y* $z_j = z_1$ *para todo* $j \leq c_0$. *Entonces:*

- $\alpha_{ij} = 0$ *para todo* $\alpha_{ij} \in \mathcal{R}(1, p-c_0+1, 2l+c_0-3)$.
- $2l \leq p-2c_0-1$.
- $z_j = z_1$ *para todo* $j \leq 2l+c_0-1$.

Supongamos que $z_j = z_1$ para todo $j \leq c_0$ y $c_0+2 \leq l$. Usando el Lema 2.6.5 y el Teorema 2.5.4 de [28] se obtiene que $2c \geq m-2l-c_0-2$ ó $c \geq m-2l-p+1$.

Teorema 2.6.6. *Supongamos que* $z_j = z_1$ *para todo* $j \leq c_0$. *Para cada* $k \geq 0$ *que satisfaga las condiciones* $2k+p-c_0-1 \leq m-2c-1$ *y* $k+c_0 \leq p$, *tenemos que* $x_1 = x_2 = \cdots = x_{k-1} ==$, *esto es,* $l \geq k$.

Notemos que el Teorema 2.6.6 es cierto cuando p no divide a $c_0+\mu$, $\mu \in [0, k-1]$.

Corolario 2.6.7. *Supongamos que* $z_j = z_1$ *para todo* $j \leq c_0$. *Entonces* $2c \geq m-2l-p+c_0-1$.

Lema 2.6.8. *Supongamos que* $2l+3 \leq m-c-1$ *y* $l \geq 3$. *Entonces o bien* $b = l+1$ *o existe el mínimo* $t \in [2, m-c-1-(2l+1)]$ *que satisface* $\alpha_{l+1,l+t} \neq 0$. *En el primer caso, tenemos que* $c \geq m-2l-p-1$, *con lo que* $2c \geq m-p-2l+c_0+1$, *y, en el otro caso, si* $2l+t+1+p-c_0-1 \leq m-2c-1$, *entonces* $z_j = z_1$ *para todo* $j \in [1, 2l]$.

A continuación se da una demostración del Teorema de Shepherd [22, Thm. 2.12].

Corolario 2.6.9. *Supongamos que* $c_0+1 \leq 2l$. *Entonces* $2c \geq m-2l-p+c_0-1$. *Consecuentemente,* $2c \geq m-2p+c_0+2$, *y si se tiene la igualdad, entonces* $c_0 = 3$ *y* $2l = p-3$.

Corolario 2.6.10. *Supongamos que* $c_0 = 2l$. *Entonces, si* $p > 7$, *tenemos que* $2c \geq m-p-2l+c_0-2$.

A continuación se estudian los casos $l = 1, 2$.

Lema 2.6.11. *Supongamos que* $2l+4 \leq m-c-1$. *Tenemos:*

1. $(y_2, y_3) \neq (0,0)$.
2. *Si* $y_3 = 0$, *entonces* $y_1 - y_2 \neq 0$.
3. *Si* $\alpha_{l+1,l+2} = 0$, *tenemos que* $z_1 = z_{2l+1} = z_{2l+2}$, $z_2 = z_{2l} = z_{2l+1}$ *para cada* $l \geq 3$.

Corolario 2.6.12. *Supongamos que* $c_0 - 1 = 2l$. *Si* $p > 7$, *entonces tenemos que* $2c \geq m-p-2l+c_0-3$.

Corolario 2.6.13. *Supongamos que* $c_0 - 2 = 2l$. *Entonces tenemos las siguientes afirmaciones:*

1. *Si* $l \geq 3$ *y* $p > 7$, *entonces* $2c \geq m-p-2l+c_0-5$.
2. *Si* $l = 2$ *y* $p > 7$, *entonces* $2c \geq m-p-1$.
3. *Si* $l = 1$ *y* $p > 7$, *entonces* $2c \geq m-p-2$.

Corolario 2.6.14.
1. *Si* $c_0 = 1$, *tenemos que* $c_0+1 \leq 2l$, *de donde* $2c \geq m-p-2l+1$.

2. *Si* $c_0 = 2$, *entonces* $c_0 \leq 2l$, *con lo que* $2c \geq m-p-2l+c_0-2 = m-p-2$ *si* $l = 1$, *y* $c_0 + 1 \leq 2l$ *si* $l \geq 2$, *luego* $2c \geq m - p - 2l + c_0 - 1$.

3. *Si* $c_0 = 3$, *entonces* $c_0 - 1 \leq 2l$, *de donde* $2c \geq m - p - 2l + c_0 - 3 = m-p-2$ *si* $l = 1$, *y* $c_0+1 \leq 2l$ *si* $l \geq 2$, *luego* $2c \geq m-p-2l+c_0-1$.

4. *Si* $c_0 = 4$, *entonces* $c_0 - 2 = 2l$, *para* $l = 1$ *y* $2c \geq m - p - 2$, *y* $c_0 = 2l$ *para* $l = 2$, *luego también* $2c \geq m - p - 2$ *y* $c_0 + 1 \leq 2l$, *para* $l \geq 3$, *con lo que* $2c \geq m - p - 2l + c_0 - 1$ *en este caso.*

En cualquier caso, para $1 \leq c_0 \leq 4$, *tenemos que*

$$2c \geq \text{mín}(m - p - 2, m - p - 2l + c_0 - 1).$$

Como aplicación de los lemas anteriores, se analizan los p-grupos de clase maximal que satisfacen la condición $c_0 = 5$.

Lema 2.6.15. *Sea* G *un* p*-grupo de clase maximal tal que* $c_0 = 5$. *Supongamos que* $p > 7$, *y* $8 + p - c_0 - 1 \leq m - 2c - 1$, *entonces* $x_1 = x_2 = x_3$.

Lema 2.6.16. *Supongamos que* $c_0 = 5$, $12 \leq m - 2c - 1$ *y* $x_1 = x_2 = x_3$. *Entonces* $x_1 = x_2 = x_3 = 0$.

Corolario 2.6.17. *Supongamos que* $c_0 = 5$ *y* $p \geq 11$. *Entonces*

$$2c \geq \text{mín}(m - p - 22, m - 2l - p + c_0 - 1).$$

En lo sucesivo, se refina el resultado anterior en los casos $2l \leq p-11$ y $p > 19$.

Lema 2.6.18. *Supongamos que* $c_0 = 5$, $p > 19$, $x_1 = x_2 = x_3$ *y* $17 \leq m - 2c - 1$, *entonces* $x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = 0$, *esto es,* $l \geq c_0 + 2$.

Teorema 2.6.19. *Supongamos que* $c_0 = 5$, $2l \leq p - 2c_0 - 1 = p - 11$ *y* $p > 19$, *entonces*

$$2c \geq m - p - 2.$$

Supongamos que $c_0 = 5$. En lo sucesivo, calculamos cotas sobre c, determinando el álgebra de Lie asociada $\mathcal{L} = \mathcal{L}(m, c, \alpha_{ij})$ en los casos $p = 13$, 17, 19, de acuerdo con los posibles valores de l en relación a m y a c. Los valores α_{ij} pueden ser dados en términos de los x_λ, que están determinados. Analizamos los posibles valores de los x_λ para $\lambda \leq p-1$ cuando la cota se alcanza: $2c = m - 2p + c_0 - 3$.

Sea t un número natural tal que $t \leq m - 2c - 1$. El sistema

$$\{f(i,j,k) = 0 \mid i + j + k \leq t\}$$

se denota por $\mathcal{S}(t)$.

Proposición 2.6.20. *Supongamos que $c_0 = 5$ y $p = 13$. Entonces $2c \geq m - 18 = m - p - 5 = m - 2p + c_0 + 3$. Más aún, se tienen las siguientes afirmaciones:*

1. *$x_1 = x_2 = x_3 = 0$ satisface $\mathcal{S}(15)$.*
2. *Si $16 \leq m - 2c - 1$, entonces se satisface $\mathcal{S}(16)$ si, y sólo si, $x_1 = x_2 = x_3 = x_4 = 0$ ó $x_1 = x_2 = x_3 = 0$, $x_4 = x_5$.*
3. *Si $17 \leq m - 2c - 1$, entonces se satisface $\mathcal{S}(17)$ si, y sólo si, $x_1 = x_2 = x_3 = x_4 = 0$.*
4. *Si $18 \leq m - 2c - 1$, entonces se satisface $\mathcal{S}(18)$ si, y sólo si, $x_1 = x_2 = x_3 = x_4 = x_5 = 0$. Por consiguiente, $2c \geq m - 18 = m - p - 5$.*

Nota 2.6.21. Concluimos que la asignación $0 = x_1 = x_2 = x_3 = x_4$, $x_6 = 8x_5$, $x_7 = 7x_5$, $x_8 = 9x_5$, $x_9 = x_5$, $x_{10} = 7x_5$, $x_{11} = 8x_5$, $x_{12} = 6x_5$ satisface $\mathcal{S}(17)$ y existe un álgebra de Lie $\mathcal{L}(m,c)$ que satisface $p = 13$ y $17 = m - 2c - 1$.

Proposición 2.6.22. *Supongamos que $c_0 = 5$ y $p = 17$. Entonces $2c \geq m - 26 = m - 2p + c_0 + 3$. Más aún, se tiene una de las siguientes afirmaciones:*

1. *Si $23 \leq m - 2c - 1$, entonces $x_i = 0$, $1 \leq i \leq 5$. Más aún, las asignaciones*

   a) *$x_1 = x_2 = x_3 = x_4 = x_5 = 0$, ó*

   b) *$x_1 = x_2 = x_3 = x_4 = 0$, $x_7 = -2x_5$, $x_6 = 2x_5$, $x_5 \neq 0$*

   satisfacen el sistema de Jacobi $\mathcal{S}(22)$.
2. *Si $25 \leq m - 2c - 1$, entonces $x_i = 0$, $1 \leq i \leq 6$, esto es, $l \geq c_0 + 2 = 7$.*

Nota 2.6.23. Concluimos que la asignación

$$0 = x_1 = x_2 = x_3 = x_4 = x_5 = x_6, x_8 = 2x_7, x_9 = -2x_7, x_{10} = 4x_7,$$

$$x_{11} = 7x_7, x_{12} = -6x_7, x_{13} = x_7, x_{14} = -8x_7, x_{15} = -7x_7, x_{16} = 8x_7,$$

satisface $\mathcal{S}(25)$ y existe un álgebr de Lie $\mathcal{L}(m,c)$ que satisface $p = 17$ y $26 = m - 2c - 1$.

Proposición 2.6.24. *Supongamos que* $c_0 = 5$ *y* $p = 19$. *Entonces* $2c \geq m - 30 = m - 2p + c_0 + 3$. *Más aún,*

1. *Si* $24 \leq m - 2c - 1$, *entonces* $x_i = 0$, $1 \leq i \leq 5$. *Además, la asignación* $x_1 = x_2 = x_3 = x_4 = 0$, $x_6 = 7x_5$, $x_7 = 4x_5$, $x_8 = 10x_5$, $x_5 \neq 0$, *satisface* $\mathcal{S}(23)$.

2. *Si* $27 \leq m - 2c - 1$, *entonces* $x_i = 0$, $1 \leq i \leq 6$, *esto es,* $l \geq c_0 + 2 = 7$. *Además, la asignación* $x_i = 0$, $1 \leq i \leq 5$, $x_7 = 7x_6$, $x_8 = 4x_6$, *satisface* $\mathcal{S}(26)$.

3. *La asignación* $x_i = 0$, $1 \leq i \leq 6$, $x_8 = -3x_7$, $x_7 \neq 0$, *satisface* $\mathcal{S}(28)$.

4. *Si* $29 \leq m - 2c - 1$, *entonces* $x_i = 0$, $1 \leq i \leq 7$.

Nota 2.6.25. Notemos que la asignación

$$0 = x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = x_7, x_9 = 7x_8, x_{10} = 4x_8, x_{11} = -9x_8,$$

$$x_{12} = -2x_8, x_{13} = -7x_8, x_{14} = 7x_8, x_{15} = -x_8,$$

$$x_{16} = 9x_8, x_{17} = 8x_8, x_{18} = -9x_8,$$

satisface $\mathcal{S}(29)$ y existe un álgebra de Lie $\mathcal{L}(m, c)$ que satisface $p = 19$, $30 = m - 2c - 1$.

Teorema 2.6.26. *Supongamos* $c_0 = 5$ *y* $p > 19$. *Entonces*

1. *Si* $2l \leq p - 11$, *entonces* $2c \geq m - p - 2$.

2. *Si* $p - 1 > 2l \geq p - 9$, *tenemos*

$$2c \geq m - 2l - p + c_0 - 1 \geq m - 2p + c_0 + 3$$

3. *Si* $2l = p - 1$, *entonces* $c \geq m - p - 1$.

Teorema 2.6.27. *Si* $c_0 = 5$, *entonces* $\mathrm{Cl}(Y_1) \leq 3$, *salvo en los siguientes casos excepcionales:*

1. $\mathrm{Cl}(Y_1) = 4$, $c = p + 4$ *y* $m \leq 4p$.

2. $c = 5$, $m \leq 2p + 2$.

Capítulo 3

Cotas para $6 \leq c_0 \leq 10$

3.1. Introducción

En este capítulo nos proponemos ampliar el estudio realizado en [29] a los p-grupos de clase maximal con $6 \leq c_0 \leq 10$.

Nuestro objetivo en este capítulo es probar el siguiente resultado, que extiende el ya conocido para $6 \leq c_0 \leq 10$:

Teorema 3.1.1. *Sea G un p-grupo de clase maximal con $p > 13$ y $6 \leq c_0 \leq 10$, entonces*

$$2c \geq \text{mín}(m - p - 2, m - 2l - p + c_0 - 1).$$

Además, para $l \geq c_0 - 2$, podemos precisar la cota de acuerdo con la tabla 3.3.

Observamos que la periodicidad módulo $p-1$ es una condición fundamental para probar este teorema, y encontramos ejemplos de álgebras de Lie que no verifican la periodicidad módulo $p-1$ pero sí satisfacen las condiciones de Jacobi (por supuesto, estas álgebras de Lie no podrán ser nunca las álgebras de Lie de p-grupos de clase maximal). En particular, encontramos un contraejemplo a la cota $2c \geq m - 2p + 5$, encontrada por G. A. Fernández-Alcober en [7] cuando se suprime la condición sobre la periodicidad módulo $p-1$.

Las técnicas computacionales desarrolladas para implementar estos lemas y obtener los resultados se explican en la sección 3.2. Los resultados teóricos preliminares necesarios ya han sido explicados en la sección 2.6.

3.2. El algoritmo

Algoritmo 3.2.1 (Configuraciones de ceros y no ceros)**.** Con este algoritmo se calculan las diferentes configuraciones de valores nulos y no nulos que pueden aparecer en el triángulo $\mathcal{T}_r$. Los calculamos mediante un proceso recursivo, empezando con la configuración para un triángulo $\mathcal{T}_1$ para una fila, correspondiente al nivel $i + j = 2$, y añadiendo una nueva fila de nivel $i+j = k+3$ en cada paso, aumentando así del nivel k al nivel $k+1$. Calculemos los $\alpha_{i,j}$ para un triángulo con r filas. Para ello, obtenemos los valores de los $\alpha_{i,j}$ cuando $i + j = r + 2$ a partir de los valores de los $\alpha_{i,j}$ para $i + j < r + 2$, teniendo en cuenta que, si r es impar, aparece una nueva variable $x_{(r+1)/2}$ y que si r es par, sabemos que $\alpha_{r/2,r/2+2} = \alpha_{r/2,r/2+1} = x_{r/2}$. Para calcular los otros elementos de la fila $i + j = r$, aplicamos la identidad de Bernoulli de derecha a izquierda.

A continuación, se analiza cuáles de los valores de la fila $i + j = r + 2$ puede anularse. Para hacerlo, consideramos una estructura de pila en la cual, para cada posición de la pila, hay una terna formada por un triángulo $\mathcal{T}_G$, un número de columna y una lista de posiciones no nulas en la matriz $\mathcal{T}_G$. Esta pila se inicializa poniendo al principio el triángulo obtenido añadiendo una nueva fila al anterior, la posición de la primera columna posible y la lista de valores no nulos en todas las filas del triángulo a excepción de la última. Para cada posible columna, empezando con la que se encuentra en la terna al final de la pila, si el elemento en dicha columna en la última fila es no nulo, es una forma lineal no nula en las variables x_i. Por tanto, podemos despejar una de dichas variables, concretamente, la que tiene coeficiente con el menor máximo divisor primo. Esta variable es substituida en el resto de las posiciones del triángulo. Si alguna de las variables que aparecen en la lista de posiciones no nulas de $\mathcal{T}_G$ se anula, seguimos con la siguiente columna. Si ninguna de ellas se anula, actuamos como sigue: Ponemos al final de la pila el nuevo triángulo formado al subsituir dicha variable, el número de la siguiente columna y la misma lista de valores no nulos, almacenamos el nuevo triángulo como una posible nueva configuración y aplicamos el algoritmo descrito aquí, hasta acabar con la última columna.

Cuando acabamos con la última columna, eliminamos la última terna del final de la pila y añadimos a la lista de valores no nulos de la última terna restante la posición correspondiente a la última fila, la columna en la que estábamos actuando (suponiendo que la lista no es vacía). Actuando de este

modo, nos aseguramos que esta columna no se puede anular de nuevo.

El algoritmo termina cuando la pila queda vacía.

Con este algoritmo recursivo, calculamos cada posible configuración de valores nulos y no nulos que puede aparecer en la última fila formada a partir de la configuración de las otras filas. Al mismo tiempo, se obtiene el mínimo primo para el cual todos los cocientes calculados tienen sentido.

Este algoritmo ha sido implmentado en lenguaje C en un PC con procesador Intel 80486 DX a 50 MHz con el sistema operativo Linux y compilador GNU C. Hemos implementado funciones para trabajar con fracciones y formas lineales con coeficientes racionales. También hemos escrito algunas funciones para despejar automáticamente una variable de una forma lineal y substituirla en otra forma. La variable aislada es la que tiene más pequeño su máximo primo divisor. Expresamos $\alpha_{i,j}$ como una forma lineal en las x_k.

Todos los posibles valores de los $\alpha_{i,j}$, con $i+j \leq r$, se guardan en un arcihvo para poderlos utilizar en un nuevo triángulo $\mathcal{T}_G$ con $r+1$ filas al mismo tiempo que vamos colocando el triángulo en la pila. El punto de partida consiste en dos triángulos de una única fila, el primero con $x_1 \neq 0$ y el segundo con $x_1 = 0$.

Ejemplo 3.2.2. Como ejemplo de aplicación de este algoritmo, podemos considerar el siguiente triángulo con 6 filas:

$$\begin{array}{ccc} -x_2 & -x_2 & x_2 \\ -2x_2 & 0 & x_2 \\ -2x_2 & x_2 & \\ -x_2 & x_2 & \\ 0 & & \\ 0 & & \end{array}$$

Ponemos este triángulo en la pila junto con la primera columna y la lista de valores no nulos (que son todos los que no hemos escrito como cero). Creamos una nueva fila. Como el nuevo triángulo debe tener 7 filas, un número impar, ponemos en la última columna (la cuarta) la nueva variable x_4. Por aplicación de la identidad de Bernoulli de derecha a izquierda, obtenemos los valores $(x_2 - x_4, -2x_2 + x_4, x_2 - x_4, x_4)$ para esta fila. El triángulo nos queda como

sigue:

$$\begin{array}{cccc} x_2 - x_4 & -2x_2 + x_4 & x_2 - x_4 & x_4 \\ -x_2 & -x_2 & x_2 & \\ -2x_2 & 0 & x_2 & \\ -2x_2 & x_2 & & \\ -x_2 & x_2 & & \\ 0 & & & \\ 0 & & & \end{array}$$

Consideremos ahora la última fila. Primeramente, $x_2 - x_4$ es una forma lineal no nula en las x_i. Podemos despejar $x_4 = x_2$. Substituimos este valor en el triángulo, y observamos que no hay ningún valor en las seis primeras filas que se anule. Por tanto, podemos escribir en un archivo este triángulo:

$$\begin{array}{cccc} 0 & -x_2 & 0 & x_2 \\ -x_2 & -x_2 & x_2 & \\ -2x_2 & 0 & x_2 & \\ -2x_2 & x_2 & & \\ -x_2 & x_2 & & \\ 0 & & & \\ 0 & & & \end{array}$$

Colocamos este triángulo al final de la pila con la segunda columna y la misma lista de valores no nulos (los que están en las seis primeras filas). Se tiene que $-x_2$ es una forma lineal no nula, podemos despejar de ella la variable x_2 como $x_2 = 0$. Pero si substituimos esta variable en las seis primeras filas, se obtiene que estos valores se anulan. La siguiente columna contiene un cero, luego la saltamos. Y la última colulmna recibe el mismo tratamiento que la segunda.

Por tanto, podemos eliminar este triángulo del final de la pila, y recuperar el anterior, pero ahora sabemos que el elemento correspondiente a $\alpha_{1,8}$ no puede anularse, esto es, $x_2 - x_4 \neq 0$.

La siguiente columna es la segunda. Tenemos la forma lineal $-2x_2 + x_4$, que es no nula. Por tanto, podemos despejar la variable x_4 de ella (su coeficiente es 1, pero el coeficiente de x_2 es 2, divisible por el primo 2) como $x_4 = 2x_2$. Podemos substituir esta variable en los otros valores, y obtenemos el triángulo

siguiente:

$$\begin{array}{cccc} -x_2 & 0 & -x_2 & 2x_2 \\ -x_2 & -x_2 & x_2 & \\ -2x_2 & 0 & x_2 & \\ -2x_2 & x_2 & & \\ -x_2 & x_2 & & \\ 0 & & & \\ 0 & & & \end{array}$$

Observemos que hay que verificar no sólo que no hay valores no nulos en las seis primeras filas que pasen a anularse, sino que tampoco se anula el elemento en la posición $\alpha_{1,8}$. Lo escribimos en un archivo, y lo ponemos al final de la pila con el número de columna 3 y la misma lista de valores no nulos. Podemos considerar ahora $-x_2$, que es una forma lineal no nula, y aislar x_2 de $-x_2 = 0$, lo que nos lleva a que $x_2 = 0$. Pero esta asignación hace que todos los valores en las seis primeras filas y en la primera posición de la última fila se anulen. Obtenemos algo similar para la última columna (hemos de considerar $p > 2$, puesto que hemos dividido entre 2).

Podemos eliminar la última terna del final de la pila, y añadir un nuevo valor no nulo para el primer triángulo, $-2x_2 + x_4$. La siguiente posibilidad es despejar una variable de $x_2 - x_4 = 0$, esto es, $x_4 = x_2$. Pero esto hace anularse al elemento $\alpha_{1,8}$, que añadimos antes a la lista de valores no nulos.

Finalmente, consideramos la última columna, y la ecuación $x_4 = 0$. Despejamos $x_4 = 0$, y substituimos este valor en los otros valores del triángulo. Obtenemos el siguiente triángulo:

$$\begin{array}{cccc} x_2 & -2x_2 & x_2 & 0 \\ -x_2 & -x_2 & x_2 & \\ -2x_2 & 0 & x_2 & \\ -2x_2 & x_2 & & \\ -x_2 & x_2 & & \\ 0 & & & \\ 0 & & & \end{array}$$

Como no hay más columnas a la derecha de ésta, no podemos substituir más variables a la derecha. De este modo, no podemos eliminar este elemento, y la pila queda vacía.

La tabla 3.1 nos muestra el número de configuraciones halladas por el algoritmo para cada c_0 con $6 \leq c_0 \leq 10$ generando triángulos con c_0+1 filas (esto

es, para $i + j \leq c_0 + 3$) y los tiempos de cómputo en un PC 486 DX/50.

Cuadro 3.1: Número de configuraciones para la generación de triángulos

c_0	Número	Tiempo (s)
6	147	6.81
7	271	10.53
8	1709	122.61
9	3800	282.87
10	30517	3026.10

Algoritmo 3.2.3 (Cálculos con z_i)**.** La segunda parte del algoritmo toma cada configuración de valores nulos y no nulos que aparecen en la primera parte. Nuestro objetivo es obtener, para la mayoría de los casos, la conclusión $z_i = z_1$ para $i \leq c_0$. Basándonos en los valores no nulos presentes en $\mathcal{T}_G$, podemos aplicar las condiciones dadas en el Lema 2.6.1 para obtener algunas igualdades entre las z_i. Para algunos de los triángulos, se obtiene directamente del mencionado Lema 2.6.1 la conclusión $z_j = z_1$ para $j \leq c_0$. Esta condición implica que $2c \geq m - p - 2$ si la aplicamos a los triángulos obtenidos en el algoritmo anterior. Para aplicar este lema, consideramos las variables $z_1, \ldots,$ z_{2c_0+2}, que son las variables que aparecen en este lema, y para cada par de valores no nulos consecutivos obtenemos la correspondiente igualdad.

Para los otros triángulos, expresamos $\alpha_{i,j}$ como función de los z_k e implementamos las condiciones $\alpha_{i,j} = 0$ para $i = j < [c_0/2] + 1$ y para los valores nulos que aparecen en el triángulo $\mathcal{T}_G$ utilizando la fórmula

$$\alpha_{i,j} = \alpha_{i,j+p-1} = \sum_{u=0}^{j+c_0-1} (-1)^u \binom{j + c_0 - 1}{u} z_{i+u}.$$

Intentamos aislar algunos de los z_j de algunas de estas igualdades y substituirlos en las otras, hasta llegar a las condiciones $z_i = z_1$ para $i \leq c_0$ o hasta que todos los $\alpha_{i,j}$, tras las substituciones, se anulan, pero no todos los z_i para $i \leq c_0$ son iguales. Estas substituciones se hance de manera similar a las dadas en el primer algoritmo, eligiendo las variables que deben ser aisladas de modo que tengan lo más pequeño posible el mayor divisor primo de su coeficiente.

Nuestro algoritmo también calcula el menor número primo p_0 que nos permite asegurar que en el cuerpo de p elementos, con $p > p_0$, dichos despejes tienen sentido (esto es, no dividimos por cero en ninguno de estos cuerpos para obtener los valores de las variables z_j o x_j).

Un nuevo problema que se nos presenta con este algoritmo es el hecho de que los enteros ordinarios de 32 bits (`long int`), entre -2^{31} y $2^{31}-1$, no son lo suficientemente grandes como para almacenar algunos de los productos y de las sumas que aparecen. Esto fuerza al ordenador a tomar algunos de los resultados módulo 2^{32}, lo que da poca confianza en el resultado. Por consiguiente, para tener enteros de más de 32 bits, creamos un nuevo tipo de datos pra enteros de más de 32 bits. En este tipo, cada entero se representa en base 2^{16} y se juntan varios enteros en una lista añadiendo también un bit de signo. De esta manera, hay que definir todas las operaciones matemáticas para esta clase de enteros y duplicar todas las funciones para que usen este nuevo tipo en lugar de los enteros "tradicionales".

La simulación por *software* de todas las operaciones matemáticas introduce una penalización de tiempo muy grande. De este modo, lo que hacemos es detectar en cada operación conflictiva (suma, producto) y en cada función que las utilice si se da un desbordamiento. En caso de desbordamiento, el cálculo se repite utilizando el nuevo tipo de datos.

Para $6 \leq c_0 \leq 10$, los únicos casos de los que no nos deshacemos cuando c_0 es impar con este algoritmo son los correspondientes al triángulo con ceros en todas las posiciones (notemos que el Lema 2.6.1 no es aplicable aquí). Para c_0 par, nos deshacemos de todos los casos menos de tres, correspondientes a $l = (c_0/2) + 1$, $l > (c_0/2) + 1$ y $x_1 \neq 0$, $x_2 = \cdots = x_{c_0/2} = 0$, $x_{(c_0/2)+1} = (-1)^{(c_0/2)+1} x_1$. La tabla 3.2 nos muestra los tiempos de cálculo.

Cuadro 3.2: Tiempos de cálculo para el segundo algoritmo

c_0	Time (s)
6	8.71
7	13.89
8	126.28
9	282.87
10	1778.70

Si queremos obtener cotas más pequeñas para estos valores, podemos modificar este algoritmo tomando más filas (por ejemplo, $c_0 + 2$ filas para obtener la cota $2c \geq m - p - 3$, en lugar de la cota $2c \geq m - p - 2$ obtenida con $c_0 + 1$ filas). Esto permite obtener $z_j = z_1$, $j \leq c_0$, en todas las configuraciones derivadas de las configuraciones en las que el primer método falla, con la excepción de aquellas con el triángulo $\mathcal{T}_G$ con sólo ceros.

A continuación mostramos un ejemplo de aplicación del algoritmo recién descrito:

Ejemplo 3.2.4. Consideremos la siguiente configuración:

$$\begin{array}{l} \bullet\ 0\ \bullet\ \bullet \\ \bullet\ \bullet\ 0 \\ 0\ \bullet\ 0 \\ \bullet\ \bullet \\ \bullet\ \bullet \\ \bullet \\ \bullet \end{array}$$

30

Por aplicación del Lema 2.6.1, obtenemos:

$$\begin{aligned} z_1 &= z_{13} \\ z_2 &= z_{14} \\ z_3 &= z_{13} \\ z_4 &= z_{13} \\ z_5 &= z_{13} \\ z_6 &= z_{13} \\ z_7 &= z_{14} \\ z_8 &= z_8 \\ z_9 &= z_{14} \\ z_{10} &= z_{13} \\ z_{11} &= z_{13} \\ z_{12} &= z_{13} \\ z_{13} &= z_{13} \\ z_{14} &= z_{14} \end{aligned}$$

Los $\alpha_{i,j}$ que se anulan son los siguientes:

$$\begin{aligned}
\alpha_{1,1} &= 5z_{13} - 5z_{14} \\
\alpha_{2,2} &= 7z_8 + 14z_{13} - 21z_{14} \\
\alpha_{3,3} &= -56z_8 - 42z_{13} + 98z_{14} \\
\alpha_{4,4} &= 126z_8 + 84z_{13} - 210z_{14} \\
\alpha_{2,7} &= 924z_8 + 658z_{13} - 1582z_{14} \\
\alpha_{3,5} &= -252z_8 - 168z_{13} + 420z_{14} \\
\alpha_{1,6} &= -330z_8 - 286z_{13} + 616z_{14} \\
\alpha_{3,4} &= -126z_8 - 84z_{13} + 210z_{14}
\end{aligned}$$

Tenemos que hacer substituciones. Teniendo en cuenta que

$$\alpha_{1,1} = 5z_{13} - 5z_{14},$$

tenemos que

$$z_{14} = z_{13}.$$

Esta substitución es válida para $p > 5$.
De este modo los valores de las $\alpha_{i,j}$ nulas y las z_i son:

$$\begin{aligned}
\alpha_{1,1} &= 0 \\
\alpha_{2,2} &= 7z_8 - 7z_{13} \\
\alpha_{3,3} &= -56z_8 + 56z_{13} \\
\alpha_{4,4} &= 126z_8 - 126z_{13} \\
\alpha_{2,7} &= 924z_8 - 924z_{13} \\
\alpha_{3,5} &= -252z_8 + 252z_{13} \\
\alpha_{1,6} &= -330z_8 + 330z_{13} \\
\alpha_{3,4} &= -126z_8 + 126z_{13}
\end{aligned}$$

$$\begin{aligned}
z_1 &= z_{13} \\
z_2 &= z_{13} \\
z_3 &= z_{13} \\
z_4 &= z_{13}
\end{aligned}$$

$$z_5 = z_{13}$$
$$z_6 = z_{13}$$
$$z_7 = z_{13}$$
$$z_8 = z_8$$
$$z_9 = z_{13}$$
$$z_{10} = z_{13}$$
$$z_{11} = z_{13}$$
$$z_{12} = z_{13}$$
$$z_{13} = z_{13}$$
$$z_{14} = z_{13}$$

Algoritmo 3.2.5 (Aplicación de la identidad de Jacobi)**.** Es posible precisar aún más las cotas $2c \geq m - 2l - p + c_0 - 1$, al menos, en los casos $c_0 - 2 \leq l$ mediante un nuevo algoritmo que usa la identidad de Jacobi. Para obtener una cota de tipo $2c \geq m - a$, suponemos que $a \leq m - 2c - 1$, y, aplicando la identidad de Jacobi, que tiene sentido mientras que $i+j+k \leq a$, se llega a la contradicción $x_l = 0$. De este modo, expresamos los valores de los $\alpha_{i,j}$ que aparecen en estas identidades de Jacobi como función de los x_k, y usamos que $x_k = 0$ para $k < l$ y que $x_l \neq 0$. Esto nos permite obtener una forma cuadrática en las variables x_i que, en algunos casos, puede ser factorizada por x_l (en algunos casos, la forma puede ser factorizado por x_{l+1} o una combinación de x_l y x_{l+1}). En consecuencia, tenemos una forma lineal en las variables x_i que se anula y podemos despejar la variable que tenga el menor número primo como máximo divisor primo de su coeficiente. Substituimos el valor de esta variable en una lista de las variables x_i, $l \leq i \leq l + c_0$. Calculamos también el menor número primo que hace que todas estas substituciones tengan sentido. Estos cálculos pueden hacerse usando el tipo entero descrito en el segundo algoritmo.

En algunos casos, la identidad de Jacobi toma la forma $ax_l^2 + bx_l x_{l+1} + cx_{l+1}^2$. En este caso, buscamos dos identidades no proporcionales de este tipo y eliminamos x_{l+1}^2 entre ellas. Esto nos permite despejar x_{l+1} como función de x_l, después de eliminar el factor común $x_l \neq 0$. Estas factorizaciones nos aparecen en el caso $l = c_0 - 2$.

De esta manera, obtenemos todas las identidades de Jacobi $f(i, j, k)$ con $i + j + k = n$ para valores crecientes de n con $n \geq 6$. El primer n para el cual obtenemos $x_l = 0$ es el a dado al principio. El algoritmo también da

los valores de x_{l+1}, ..., x_{l+c_0} que satisfacen todas las ecuaciones de Jacobi $f(i,j,k)$ para $i+j+k<a$.

Para saber los primos para los cuales este argumento es válido, tenemos que recordar la periodicidad módulo $p-1$, que nos permite sólo asignar hasta $(p-3)/2$ variables (las otras están asignadas por la periodicidad).

Podemos obtener también una nueva versión de este programa en el caso en que fijamos un primo p. Podemos argumentar como antes con enteros ordinarios módulo p (lo cual incrementa considerablemente la velocidad de los cálculos). La periodicidad módulo $p-1$ está implementada mediante las condiciones $\alpha_{1,p}=0$ y $\alpha_{i,p+i}=x_i$; cada una de estas condiciones nos da una nueva variable, hasta que obtenemos todos los valores de x_i, $i\leq p-2$.

Describimos a continuación algunos datos sobre esta última versión del algoritmo. Algo que resulta fundamental en el desarrollo de este algoritmo es un algoritmo de factorización de formas cuadráticas en $\mathbb{Z}_p$, que esquematizamos como sigue:

Supongamos que la forma que deseamos factorizar viene dada por $\sum a_{i,j}x_ix_j$, en primer lugar se estudia si existe un $a_{i,i}$ distinto de cero o si todos los $a_{i,i}$ valen cero.

1. Si existe un i tal que $a_{i,i}=0$, en la descomposición de la forma bilineal en factores lineales debe aparecer la variable x_i. Entonces la descomposición es del tipo $(x_i+f)(x_i+g)$. Para lograr el resto de los sumandos, nos fijamos en los términos en x_ix_j con $j\neq i$. Tomemos el primero de ellos. Si la forma factorizase, en uno de los factores o en ambos debería aparecer x_j. Para obtener el coeficiente de x_j en estos factores se tiene sólo en cuenta la parte $a_{i,i}x_i^2+a_{i,j}x_ix_j+a_{j,j}x_j^2$ y se escribe en la forma $(x_i+bx_j)(x_i+cx_j)$, siendo b y c las raíces de la ecuación $a_{i,i}y^2+a_{i,j}y+a_{j,j}=0$. Si esta ecuación en y no tiene raíces, la descomposición no es posible y se termina el proceso. Si tiene raíces, ya tenemos los dos primeros sumandos de la factorización. Para buscar el siguiente sumando, tomamos el siguiente x_ix_k que aparece en la forma bilineal con $k\neq i$, $a_{i,k}\neq 0$. Repetimos el estudio de $a_{i,i}x_i^2+a_{i,k}x_ix_k+a_{k,k}x_k^2$. Si la ecuación en y asociada tiene solución y sus raíces son d y e, entonces los primeros sumandos serán $(x_i+bx_j+dx_k)(x_i+cx_j+ex_k)$ o $(x_i+bx_j+ex_k)(x_i+cx_j+dx_k)$ y el propio programa se encarga de seleccionar la factorización adecuada o de determinar que la factorización no es posible por no verificarse las relaciones debidas entre x_j y

x_k. Dicho procedimiento se sigue hasta que se obtiene la factorización o se llega a la conclusión de que ésta es imposible.

2. Si todos los coeficientes $a_{i,i}$ son nulos, entonces x_i sólo puede aparecer en uno de los factores. Entonces, si es que existe descomposición, debe ser de la forma $(\sum b_i x_i)(\sum c_i x_i)$, donde $b_i = 0$ si $c_i \neq 0$. Se toma un $x_i x_j$ tal que $a_{i,j} \neq 0$. Se examinan las relaciones que deben existir entre los coeficientes y así se obtiene la descomposición o se determina que ésta no existe.

En este algoritmo es necesario resolver una ecuación cuadrática en $\mathbb{Z}_p$. Los algoritmos para su resolución, en particular, el algoritmo de Shanks para la extracción de raíces cuadradas en $\mathbb{Z}_p$, figuran en [9, capítulos 3, 4 y 11].

El algoritmo de cálculo de cotas comienza expresando x_k para $k \geq (p-1)/2$ en función de x_l, $x_{l+1}, \ldots, x_{(p-3)/2}$ utilizando la periodicidad módulo $p-1$. Comenzamos con $n = 6$, $n_c = 5$, $n_m = 5$. Calculamos las expresiones de Jacobi en el nivel n, y analizamos si alguna de ellas factoriza como producto de dos formas lineales. Si es así, guardamos uno de ellos en una pila junto con los valores actuales de n_c y las x_i y despejamos de la otra una de las variables, substituimos y volvemos a plantear las ecuaciones de Jacobi para el nivel $n = n_c + 1$. Si ninguna de las formas factoriza, entonces aumentamos n y calculamos las expresiones de Jacobi para el nuevo valor de n hasta que alguna forma factorice. Al mismo tiempo, el valor de n_m se incrementa si n pasa a ser mayor que n_m. Este proceso se sigue realizando hasta que se llegue a la contradicción $x_l = 0$. En este caso, se recupera de la pila el valor del factor que nos quedaba y despejamos una variable de esta forma y volvemos a plantear las ecuaciones de Jacobi para el nivel $n = n_c + 1$, con el n_c tomado de la pila y los valores de las x_i almacenados en la pila. Este proceso se reitera hasta que tengamos la pila vacía.

Desde luego, hay que tener en cuenta que si uno de los factores que aparece en una factorización es x_l, o si ambos factores son iguales, no hace falta guardar en la pila el otro factor, lo cual permite no considerar algunas ramas en el algoritmo.

Se puede obtener una cota "exacta" si la contradicción se alcanza en el nivel máximo n_m. Sin embargo, en algunos casos esto no ocurre, ya que no aparece ninguna ecuación de Jacobi que factorice hasta un determinado nivel n_m y después, al hacer las substituciones en niveles n más bajos, se obtenga la

contradicción. Esto ocurre para valores de l pequeños, como se ve en las tablas 4.26 y 4.27.

Se puede instruir al algoritmo para que dé los valores de las variables x_i en cada nivel, en particular, en el nivel anterior al n_m de la contradicción.

La versión de este algoritmo sin números primos nos llevó $1081,44$ segundos de CPU, mientras que la segunda versión necesitó 781,01 segundos de CPU. Escribimos en las tablas 3.3 las cotas obtenidas de modo que el número a significa $2c \geq m - a$.

Los valores para las x_i que nos permiten afirmar que estas cotas son las mejores posibles son los que aparecen en las tablas 3.4 a 3.8. El signo $*$ denota que cualquier valor posible de la variable en dicha columna satisface el álgebra de Lie correspondiente, mientras que combinaciones que no son de la forma $x_i = r \cdot x_l$ aparecen escritas en la forma $x_i = r_1 x_{j_1} + r_2 x_{j_2} + \cdots$.

El siguiente ejemplo nos muestra el resultado del algoritmo para $c_0 = 6$.

Ejemplo 3.2.6. Supongamos que $l = 5$, $c_0 = 6$.

En el nivel 12:

Consideremos la ecuación de Jacobi: $f(1,5,6) = x_5\big(-330x_5 + 252x_6 - 84x_7 + 8x_8\big)$.

Despejamos x_8 de

$$-330x_5 + 252x_6 - 84x_7 + 8x_8 = 0.$$

La substitución es válida para $p > 2$.

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= x_6 \\ x_7 &= x_7 \\ x_8 &= (165/4)x_5 - (63/2)x_6 + (21/2)x_7 \\ x_9 &= x_9 \\ x_{10} &= x_{10} \\ x_{11} &= x_{11} \end{aligned}$$

En el nivel 13:

Consideremos la ecuación de Jacobi: $f(2,5,6) = x_5\big(-990x_5 + 672x_6 - 168x_7 + x_9\big)$.

Cuadro 3.3: Cotas obtenidas con la identidad de Jacobi

$c_0 = 6$	$p = 17$	$p = 19$	$p = 23$	$p \geq 29$
$l = 4$	20	21	16	16
$l = 5$	22	24	17	18
$l = 6$	23	26	29	20
$l = 7$	25	27	32	22

$c_0 = 7$	$p = 17$	$p = 19$	$p = 23$	$p = 29$	$p = 31$	$p > 31$
$l = 5$	21	23	26	19	19	19
$l = 6$	23	25	29	21	21	21
$l = 7$		27	31	22	23	23
$l = 8$		28	33	38	24	25

$c_0 = 8$	$p = 19$	$p = 23$	$p = 29$	$p = 31$	$p > 31$
$l = 6$	24	28	21	22	22
$l = 7$	25	30	35	23	24
$l = 8$		32	38	39	26
$l = 9$		33	40	42	28

$c_0 = 9$	$p = 19$	$p = 23$	$p = 29$	$p = 31$	$p = 37$	$p > 37$
$l = 7$	24	29	35	36	25	25
$l = 8$		31	37	39	27	27
$l = 9$		32	39	41	28	29
$l = 10$			41	43	48	31

$c_0 = 10$	$p = 23$	$p = 29$	$p = 31$	$p = 37$	$p = 41$	$p = 43$	$p > 43$
$l = 8$	29	36	38	27	28	28	28
$l = 9$	31	38	40	45	30	30	30
$l = 10$		40	42	48	31	32	32
$l = 11$		41	44	50	53	33	34

Cuadro 3.4: Álgebras de Lie para $c_0 = 6$

$l = 4$	p	x_5/x_4	x_6/x_4	x_7/x_4	x_8/x_4	x_9/x_4	x_{10}/x_4
	17	2	15	4			
	19	7	4	10	17		
	> 23	20/11	25/11	200/77	35/11	56/11	210/11

$l = 5$	p	x_6/x_5	x_7/x_5	x_8/x_5	x_9/x_5	x_{10}/x_5	x_{11}/x_5
	17	10	14				
	19	7	4	10			
	23	14	8	12	10	21	
	> 23	5/2	25/6	25/4	10	21	105

$l = 6$	p	x_7/x_6	x_8/x_6	x_9/x_6	x_{10}/x_6	x_{11}/x_6	x_{12}/x_6
	17	$*$					
	19	16	10				
	23	20	7	21	4		
	> 23	36/11	7	40/3	27	72	462

$l = 7$	p	x_8/x_7	x_9/x_7	x_{10}/x_7	x_{11}/x_7	x_{12}/x_7	x_{13}/x_7
	17						
	19	$*$					
	23	20	7	21			
	> 23	91/22	364/33	26	65	429/2	1716

Cuadro 3.5: Álgebras de Lie para $c_0 = 7$

$l = 5$	p	x_6/x_5	x_7/x_5	x_8/x_5	x_9/x_5	x_{10}/x_5	x_{11}/x_5	x_{12}/x_5
	17	12	7					
	19	16	10	8				
	23	20	7	21	4	3		
	> 23	30/13	45/13	175/39	75/13	108/13	210/13	990/13

$l = 6$	p	x_7/x_6	x_8/x_6	x_9/x_6	x_{10}/x_6	x_{11}/x_6	x_{12}/x_6	x_{13}/x_6
	17	13						
	19	12	14					
	23	20	7	21	4			
	> 23	3	63/11	28/3	15	27	66	396

$l = 7$	p	x_8/x_7	x_9/x_7	x_{10}/x_7	x_{11}/x_7	x_{12}/x_7	x_{13}/x_7	x_{14}/x_7
	19	13						
	23	8	20	6				
	29	6	1	2	6	19	28	
	> 29	49/13	98/11	196/11	35	77	231	1716

$l = 8$	p	x_9/x_8	x_{10}/x_8	x_{11}/x_8	x_{12}/x_8	x_{13}/x_8	x_{14}/x_8	x_{15}/x_8
	19							
	23	11	19					
	29	18	12	16	17	24		
	31	7	13	29	13	12	2	
	> 31	60/13	1890/143	350/11	75	198	715	6435

Cuadro 3.6: Álgebras de Lie para $c_0 = 8$

$l = 6$	p	x_7/x_6	x_8/x_6	x_9/x_6	x_{10}/x_6	x_{11}/x_6	x_{12}/x_6	x_{13}/x_6	x_{14}/x_6
	19	13	8						
	23	8	20	6	5				
	29	26	2	24	4	14	26	18	
	> 29	14/5	49/10	392/55	49/5	14	231/10	264/5	3003/10

$l = 7$	p	x_8/x_7	x_9/x_7	x_{10}/x_7	x_{11}/x_7	x_{12}/x_7	x_{13}/x_7	x_{14}/x_7	x_{15}/x_7
	19	$*$							
	23	11	19	3					
	29	18	12	16	17	24	19		
	31	19	29	19	11	23	15	13	
	> 31	7/2	98/13	147/11	245/11	77/2	77	429/2	3003/2

$l = 8$	p	x_9/x_8	x_{10}/x_8	x_{11}/x_8	x_{12}/x_8	x_{13}/x_8	x_{14}/x_8	x_{15}/x_8	x_{16}/x_8
	23	12	8						
	29	18	12	16	17	24			
	31	27	23	30	26	3	18		
	> 31	64/15	144/13	3360/143	140/3	96	1144/5	2288/3	6435

$l = 9$	p	x_{10}/x_9	x_{11}/x_9	x_{12}/x_9	x_{13}/x_9	x_{14}/x_9	x_{15}/x_9	x_{16}/x_9	x_{17}/x_9
	23	$*$							
	29	3	3	10	9				
	31	27	23	30	26	3			
	> 31	51/10	204/13	510/13	1190/13	221	3094/5	2431	24310

Cuadro 3.7: Álgebras de Lie para $c_0 = 9$

$l=7$	p	x_8/x_7	x_9/x_7	x_{10}/x_7	x_{11}/x_7	x_{12}/x_7	x_{13}/x_7	x_{14}/x_7	x_{15}/x_7	x_{16}/x_7
	19	$*$								
	23	$*$	$x_9 = x_7 + 14x_8$	$x_{10} = 21x_7 + 3x_8$						
	29	18	12	16	17	24	19			
	31	27	23	30	26	3	18	29		
	> 31	56/17	112/17	2352/221	2940/187	392/17	616/17	1144/17	3003/17	20020/17

$l=8$	p	x_9/x_8	x_{10}/x_8	x_{11}/x_8	x_{12}/x_8	x_{13}/x_8	x_{14}/x_8	x_{15}/x_8	x_{16}/x_8	x_{17}/x_8
	23	17	12							
	29	3	3	10	9	1				
	31	27	23	30	26	3	18			
	> 31	4	48/5	240/13	420/13	56	104	1144/5	715	5720

$l=9$	p	x_{10}/x_9	x_{11}/x_9	x_{12}/x_9	x_{13}/x_9	x_{14}/x_9	x_{15}/x_9	x_{16}/x_9	x_{17}/x_9	x_{18}/x_9
	23	$*$								
	29	7	27	6	19					
	31	11	25	3	11	21				
	37	20	32	2	31	15	14	36	21	
	> 37	81/17	27/2	396/13	810/13	126	273	702	2574	24310

$l=10$	p	x_{11}/x_{10}	x_{12}/x_{10}	x_{13}/x_{10}	x_{14}/x_{10}	x_{15}/x_{10}	x_{16}/x_{10}	x_{17}/x_{10}	x_{18}/x_{10}	x_{19}/x_{10}
	29	$*$	$x_{12} = 17x_{10} + 24x_{11}$	$x_{13} = 17x_{10} + 3x_{11}$	$x_{14} = 13x_{10} + 16x_{11}$					
	31	5	8	29	4					
	37	23	13	34	3	7	36	15		
	> 37	95/17	627/34	627/13	114	266	665	1976	8398	92378

Despejamos x_9 a partir de

$$-990x_5 + 672x_6 - 168x_7 + x_9 = 0.$$

La substitución es válida para $p > 1$.

$$\begin{aligned}
x_5 &= x_5 \\
x_6 &= x_6 \\
x_7 &= x_7 \\
x_8 &= (165/4)x_5 - (63/2)x_6 + (21/2)x_7 \\
x_9 &= 990x_5 - 672x_6 + 168x_7 \\
x_{10} &= x_{10} \\
x_{11} &= x_{11}
\end{aligned}$$

En el nivel 14:

Consideremos la ecuación de Jacobi: $f(3,5,6) = x_5\big(-4675x_5 + 3060x_6 - 714x_7\big)$.

Cuadro 3.8: Álgebras de Lie para $c_0 = 10$

$l = 8$	p	x_9/x_8	x_{10}/x_8	x_{11}/x_8	x_{12}/x_8	x_{13}/x_8	x_{14}/x_8	x_{15}/x_8	x_{16}/x_8	x_{17}/x_8	x_{18}/x_8
	23	$*$	$*$								
	29	7	27	6	19	26					
	31	11	25	3	11	21	1				
	37	33	28	21	23	21	1	7	26	14	
	> 37	72/19	162/19	288/19	5940/247	9072/247	1092/19	1872/19	3861/19	11440/19	87156/19

$l = 9$	p	x_{10}/x_9	x_{11}/x_9	x_{12}/x_9	x_{13}/x_9	x_{14}/x_9	x_{15}/x_9	x_{16}/x_9	x_{17}/x_9	x_{18}/x_9	x_{19}/x_9
	23	$*$									
	29	$*$	$x_{11} = 17x_9 + 24x_{10}$		$x_{12} = 17x_9 + 3x_{10}$						
	31	5	8	29	4	30					
	37	23	13	34	3	7	36	15	36		
	> 37	9/2	405/34	99/4	594/13	81	147	585/2	702	2431	21879

$l = 10$	p	x_{11}/x_{10}	x_{12}/x_{10}	x_{13}/x_{10}	x_{14}/x_{10}	x_{15}/x_{10}	x_{16}/x_{10}	x_{17}/x_{10}	x_{18}/x_{10}	x_{19}/x_{10}	x_{20}/x_{10}
	29	25	9	11							
	31	$*$	$x_{12} = 4x_{10} + 12x_{11}$		$x_{13} = 26x_{10} + 8x_{11}$						
	37	23	13	34	3	7	36	15			
	41	29	21	34	21	4	22	21	37	25	
	> 41	100/19	275/17	660/17	165/2	168	350	800	2210	8840	92378

$l = 11$	p	x_{12}/x_{11}	x_{13}/x_{11}	x_{14}/x_{11}	x_{15}/x_{11}	x_{16}/x_{11}	x_{17}/x_{11}	x_{18}/x_{11}	x_{19}/x_{11}	x_{20}/x_{11}	x_{21}/x_{11}
	29	$*$	$*$								
	31	12	24	14							
	37	4	20	9	23	24	14				
	41	5	8	1	4	13	5	31	30		
	43	14	14	26	20	19	10	19	19	24	
	> 43	231/38	6930/323	1001/17	4851/34	1323/4	784	2040	6426	29393	352716

Despejamos x_7 a partir de

$$-4675x_5 + 3060x_6 - 714x_7 = 0.$$

La substitución es válida para $p > 17$.

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= x_6 \\ x_7 &= -(275/42)x_5 + (30/7)x_6 \\ x_8 &= -(55/2)x_5 + (27/2)x_6 \\ x_9 &= -110x_5 + 48x_6 \\ x_{10} &= x_{10} \\ x_{11} &= x_{11} \end{aligned}$$

En el nivel 15:
Consideremos la ecuación de Jacobi: $f(4,5,6) = x_5\big(-924x_5 + 378x_6 - x_{10}\big)$.
Despejamos x_{10} a partir de

$$-924x_5 + 378x_6 - x_{10} = 0.$$

La substitución es válida para $p > 1$.

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= x_6 \\ x_7 &= -(275/42)x_5 + (30/7)x_6 \\ x_8 &= -(55/2)x_5 + (27/2)x_6 \\ x_9 &= -110x_5 + 48x_6 \\ x_{10} &= -924x_5 + 378x_6 \\ x_{11} &= x_{11} \end{aligned}$$

En el nivel 16:
Consideremos la ecuación de Jacobi:

$$f(3,6,7) = x_6\big(-(20995/7)x_5 + (8398/7)x_6\big).$$

Despejamos x_6 a partir de

$$-(20995/7)x_5 + (8398/7)x_6 = 0.$$

La substitución es válida para $p > 19$.

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= (5/2)x_5 \\ x_7 &= (25/6)x_5 \\ x_8 &= (25/4)x_5 \\ x_9 &= 10x_5 \\ x_{10} &= 21x_5 \\ x_{11} &= x_{11} \end{aligned}$$

En el nivel 17:
Consideremos la ecuación de Jacobi:

$$f(4,6,7) = x_5\big(-(735/2)x_5 + (7/2)x_{11}\big).$$

Despejamos x_{11} a partir de

$$-(735/2)x_5 + (7/2)x_{11} = 0.$$

La substitución es válida para $p > 7$.

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= (5/2)x_5 \\ x_7 &= (25/6)x_5 \\ x_8 &= (25/4)x_5 \\ x_9 &= 10x_5 \\ x_{10} &= 21x_5 \\ x_{11} &= 105x_5 \end{aligned}$$

Al nivel 18:
Consideremos la ecuación de Jacobi: $f(5,6,7) = x_5\big(-(4845/4)x_5\big)$.
Despejamos $x_5 = 0$.
The substitution is valid for $p > 19$.

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= (5/2)x_5 \end{aligned}$$

$$\begin{aligned} x_7 &= (25/6)x_5 \\ x_8 &= (25/4)x_5 \\ x_9 &= 10x_5 \\ x_{10} &= 21x_5 \\ x_{11} &= 105x_5 \end{aligned}$$

Recordemos que, por la multiplicidad módulo $p-1$, obtenemos que $p \geq 25$.
Consideremos ahora los valores $l = 5$, $c_0 = 6$, $p = 17$.
Éstos son los valores iniciales:

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= x_6 \\ x_7 &= x_7 \\ x_8 &= 3x_5 + 11x_6 + 2x_7 \\ x_9 &= 4x_5 + 8x_6 + 15x_7 \\ x_{10} &= 4x_5 + 16x_6 + 4x_7 \\ x_{11} &= 4x_5 + 14x_6 + 7x_7 \\ x_{12} &= 5x_5 + 13x_6 + 11x_7 \\ x_{13} &= 15x_5 + 7x_6 + x_7 \\ x_{14} &= 8x_5 + 9x_6 + 9x_7 \\ x_{15} &= 10x_5 + 5x_6 + 10x_7 \end{aligned}$$

En el nivel 21:
Consideremos la ecuación de Jacobi: $f(6,7,8) = x_5\big(2x_6 + x_7\big)$.
Despejamos x_7 a partir de

$$2x_6 + x_7 = 0.$$

Consideremos la ecuación de Jacobi: $f(4,8,9) = x_5\big(8x_5 + 6x_6\big)$.
Despejamos x_6 a partir de

$$8x_5 + 6x_6 = 0.$$

$$x_5 = x_5$$

$$x_6 = 10x_5$$
$$x_7 = 14x_5$$

En el nivel 22
Consideremos la ecuación de Jacobi: $f(5,8,9) = x_5\big(15x_5\big)$.
Despejamos x_5 a partir de

$$15x_5 = 0.$$

$$x_5 = 0$$
$$x_6 = 10x_5$$
$$x_7 = 14x_5$$

Consideremos ahora los valores $l = 5$, $c_0 = 6$, $p = 19$.
Éstos son los valores iniciales:

$$\begin{aligned}
x_5 &= x_5\\
x_6 &= x_6\\
x_7 &= x_7\\
x_8 &= x_8\\
x_9 &= 5x_5 + 12x_6 + 12x_7 + 7x_8\\
x_{10} &= 10x_5 + 13x_6 + x_7 + 4x_8\\
x_{11} &= 17x_5 + x_6 + 4x_7 + 10x_8\\
x_{12} &= x_5 + 4x_6 + 7x_7 + 17x_8\\
x_{13} &= 16x_5 + 15x_6 + 18x_7 + 12x_8\\
x_{14} &= 6x_5 + 8x_6 + 7x_7 + 7x_8\\
x_{15} &= 10x_5 + 14x_6 + 16x_7 + 18x_8\\
x_{16} &= 11x_5 + 17x_6 + 2x_7 + 9x_8\\
x_{17} &= 12x_5 + 5x_6 + 11x_7 + 8x_8
\end{aligned}$$

En el nivel 12:
Consideremos la ecuación de Jacobi: $f(1,5,6) = x_5\big(12x_5+5x_6+11x_7+8x_8\big)$.

Despejamos x_8 a partir de

$$12x_5 + 5x_6 + 11x_7 + 8x_8 = 0.$$

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= x_6 \\ x_7 &= x_7 \\ x_8 &= 8x_5 + 16x_6 + x_7 \end{aligned}$$

En el nivel 13:
Consideremos la ecuación de Jacobi: $f(2,5,6) = x_5\big(2x_5 + 17x_6 + 3x_7\big)$.
Despejamos x_7 a partir de

$$2x_5 + 17x_6 + 3x_7 = 0.$$

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= x_6 \\ x_7 &= 12x_5 + 7x_6 \\ x_8 &= x_5 + 4x_6 \end{aligned}$$

En el nivel 23:
Consideremos la ecuación de Jacobi: $f(6,8,9) = x_5\big(13x_5 + 9x_6\big)$.
Despejamos x_6 a partir de

$$13x_5 + 9x_6 = 0.$$

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= 7x_5 \end{aligned}$$

$$x_7 = 4x_5$$
$$x_8 = 10x_5$$

En el nivel 24:
Consideremos la ecuación de Jacobi: $f(7,8,9) = x_5\big(11x_5\big)$.
Despejamos x_5 a partir de

$$11x_5 = 0.$$

$$x_5 = 0$$
$$x_6 = 7x_5$$
$$x_7 = 4x_5$$
$$x_8 = 10x_5$$

Consideremos ahora los valores $l = 5$, $c_0 = 6$, $p = 23$.
Éstos son los valores iniciales:

$$\begin{aligned}
x_5 &= x_5\\
x_6 &= x_6\\
x_7 &= x_7\\
x_8 &= x_8\\
x_9 &= x_9\\
x_{10} &= x_{10}\\
x_{11} &= 22x_5 + 4x_6 + 5x_7 + 13x_8 + 21x_9 + 20x_{10}\\
x_{12} &= 4x_5 + 10x_6 + 12x_7 + 12x_8 + 19x_9 + 7x_{10}\\
x_{13} &= 10x_5 + 9x_6 + 22x_7 + 11x_8 + 21x_9 + 21x_{10}\\
x_{14} &= x_5 + 2x_6 + 22x_7 + 19x_8 + 15x_9 + 4x_{10}\\
x_{15} &= 3x_5 + 17x_6 + 22x_7 + 5x_8 + 11x_9 + 3x_{10}\\
x_{16} &= 20x_5 + 15x_6 + 9x_7 + 15x_8 + 22x_9 + 2x_{10}\\
x_{17} &= 14x_5 + 5x_6 + 2x_7 + 12x_8 + 11x_9 + 16x_{10}\\
x_{18} &= 20x_5 + 9x_6 + 16x_7 + 3x_8 + 3x_9 + 9x_{10}
\end{aligned}$$

$$x_{19} = 4x_5 + 10x_6 + 8x_7 + 18x_8 + 8x_9 + 22x_{10}$$
$$x_{20} = 3x_5 + 5x_7 + 18x_8 + 20x_9 + 11x_{10}$$
$$x_{21} = 15x_5 + x_6 + 9x_7 + 10x_8 + 19x_9 + 10x_{10}$$

En el nivel 12:
Consideremos la ecuación de Jacobi: $f(1,5,6) = x_5\big(15x_5 + 22x_6 + 8x_7 + 8x_8\big)$.
Despejamos x_8 a partir de

$$15x_5 + 22x_6 + 8x_7 + 8x_8 = 0.$$

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= x_6 \\ x_7 &= x_7 \\ x_8 &= x_5 + 3x_6 + 22x_7 \\ x_9 &= x_9 \\ x_{10} &= x_{10} \end{aligned}$$

En el nivel 13:
Consideremos la ecuación de Jacobi: $f(2,5,6) = x_5\big(22x_5 + 5x_6 + 16x_7 + x_9\big)$.
Despejamos x_9 a partir de

$$22x_5 + 5x_6 + 16x_7 + x_9 = 0.$$

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= x_6 \\ x_7 &= x_7 \\ x_8 &= x_5 + 3x_6 + 22x_7 \\ x_9 &= x_5 + 18x_6 + 7x_7 \\ x_{10} &= x_{10} \end{aligned}$$

En el nivel 14:
Consideremos la ecuación de Jacobi: $f(3,5,6) = x_5\big(17x_5 + x_6 + 22x_7\big)$.
Despejamos x_7 a partir de

$$17x_5 + x_6 + 22x_7 = 0.$$

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= x_6 \\ x_7 &= 17x_5 + x_6 \\ x_8 &= 7x_5 + 2x_6 \\ x_9 &= 5x_5 + 2x_6 \\ x_{10} &= x_{10} \end{aligned}$$

En el nivel 15:
Consideremos la ecuación de Jacobi: $f(4,5,6) = x_5\big(19x_5 + 10x_6 + 22x_{10}\big)$.
Despejamos x_{10} a partir de

$$19x_5 + 10x_6 + 22x_{10} = 0.$$

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= x_6 \\ x_7 &= 17x_5 + x_6 \\ x_8 &= 7x_5 + 2x_6 \\ x_9 &= 5x_5 + 2x_6 \\ x_{10} &= 19x_5 + 10x_6 \end{aligned}$$

En el nivel 16:
Consideremos la ecuación de Jacobi: $f(3,6,7) = x_6\big(17x_5 + 7x_6\big)$.

Despejamos x_6 a partir de

$$17x_5 + 7x_6 = 0.$$

$$\begin{aligned} x_5 &= x_5 \\ x_6 &= 14x_5 \\ x_7 &= 8x_5 \\ x_8 &= 12x_5 \\ x_9 &= 10x_5 \\ x_{10} &= 21x_5 \end{aligned}$$

En el nivel 17:
Consideremos la ecuación de Jacobi: $f(4,6,7) = x_5\big(2x_5\big)$.
Despejamos x_5 a partir de

$$2x_5 = 0.$$

$$\begin{aligned} x_5 &= 0 \\ x_6 &= 14x_5 \\ x_7 &= 8x_5 \\ x_8 &= 12x_5 \\ x_9 &= 10x_5 \\ x_{10} &= 21x_5 \end{aligned}$$

Nota 3.2.7. Ya mencionamos anteriormente que existían algunos casos en los cuales la identidad $z_i = z_j$ para $j \leq c_0$ no se da, entre ellos, los correspondientes a $x_1 \neq 0$, $x_2 = x_3 = \cdots = x_{c/2} = 0$, $x_{(c/2)+1} = (-1)^{(c/2)+1}x_1$. Para estas configuraciones, obtenemos la igualdad $z_1 = z_j$ cuando tomamos una fila más. Esto nos permite afirmar que $2c \geq m - p - 3$, pero podemos usar argumentos similares a los empleados para $l \geq c_0 - 2$ y obtener así cotas mejores. La tabla 3.9 nos muestra las cotas correspondientes a $c_0 = 6$, $c_0 = 8$ y $c_0 = 10$, así como las correspondientes álgebras de Lie.

Cuadro 3.9: Álgebras de Lie correspondientes a los casos especiales con c_0 par

c_0	6	8	10
$2c \geq$	12	14	16
p	> 7	> 13	> 11
x_2	0	0	0
x_3	0	0	0
x_4	x_1	0	0
x_5	$2x_1$	$-x_1$	0
x_6	$3x_1$	$(-10/3)x_1$	x_1
x_7	$(32/7)x_1$	$(-73/9)x_1$	$5x_1$
x_8	$9x_1$	$(-487/27)x_1$	$18x_1$
x_9		$(-3124/81)x_1$	$59x_1$
x_{10}		$16x_1$	$188x_1$
x_{11}			$(6540/11)x_1$
x_{12}			$25x_1$

Cuadro 3.10: Los otros casos

c_0	8	9	10	10
l	5	6	6	7
$2c \geq$	$p+3$	$p+4$	$p+3$	$p+5$

Los casos no desechados por nuestros algoritmos se estudian añadiendo suficientes filas al triángulo $\mathcal{T}_G$ hasta que obtenemos la condición $z_1 = z_j$ para $j \leq c_0$. La tabla 3.10 muestra las cotas halladas para estos casos.

Nota 3.2.8. Existen álgebras de Lie no asociativas sobre $\mathbb{F}_p$ que no cumplen la periodicidad módulo $p-1$. En efecto, consideremos el álgebra de Lie dada por $x_1 = x_2 = x_3 = 0$, $x_4 = x_5 \neq 0$, $x_6 = -4x_4$, $x_7 = -28x_4$, $x_8 = 3x_4$, $x_9 = 4x_4$, $x_{10} = -10x_4$, $x_{11} = 8x_4$, $x_{12} = x_{13} = 0$, $x_{14} \neq 0$, $x_{15} = x_{16} = x_{17} = x_{18} = x_{19} = x_{20} = 0$, pero $x_4 \neq x_{4+17-1} = x_{20} = 0$. Esta álgebra satisface las igualdades de Jacobi, pero no verifica la propiedad **(C4)** (periodicidad módulo $p-1$), luego no puede ser el álgebra de Lie de un p-grupo. Más aún, no podemos omitir la hipótesis **(C4)** para probar la cota $2c \geq m - 2p + 5$ dada por Fernández-Alcober en [7], porque $m-2c-1 \geq 34$ para esta álgebra.

Capítulo 4

Nuevas cotas

Tras aplicar los algoritmos mostrados en el capítulo 3 conjeturamos la existencia de regiones dependientes de c_0, l y p en las cuales las cotas para el grado de conmutatividad de un p-grupo de clase maximal son exactas, en el sentido de la existencia de álgebras de Lie para los niveles anteriores. Mediante los invariantes l y c_0 asociados a estos grupos, encontramos nuevas cotas para $c(G)$ de la forma $2c \geq m - g(c_0, l, p)$ para $g(c_0, l, p)$ una función adecuada de c_0, l y p. Presentamos algunas conjeturas para estas cotas, y probamos algunas de ellas. En muchos de estos resultados juegan un papel importante los coeficientes $z_i = \alpha_{i,p-c_0}$, es éste el motivo de la siguiente sección.

4.1. Resultados obtenidos a partir de las z_i

Lema 4.1.1. *Supongamos que* $2l + c_0 \leq p \leq 2c_0 + 2l - 3$ *y* $z_1 = z_j$ *para* $1 \leq j \leq p - c_0 - 2l + 3$. *Entonces se llega a la contradicción* $x_l = 0$.

Demostración. Como $z_1 = z_j$ for $1 \leq l \leq p - c_0 - 2l + 3$, obtenemos que $\alpha_{i,p-c_0+1} = 0$ para $1 \leq i \leq p - c_0 - 2l + 2$. Tenemos también que

$$\begin{aligned}\alpha_{i,p-c_0+1} &= \sum_{k=i}^{[(p-c_0+i)/2]} (-1)^{k-i} \binom{p - c_0 - k}{k - i} x_k \\ &= \sum_{s=1}^{[(p-c_0+i)/2]-l+1} (-1)^{s+l-1-i} \binom{p - c_0 - l + 1 - s}{p - c_0 - 2l + 2 - 2s + i} x_{s+l-1}.\end{aligned}$$

Denotemos $r = p - c_0 - l + 1$, $t = e = p - c_0 - 2l + 2$. Si el determinante de la matriz de coeficientes es un múltiplo de p (notemos que es una matriz cuadrada, porque la condición

$$\left[\frac{p - c_0 + e}{2}\right] - l + 1 = \left[\frac{p - c_0 + p - c_0 - 2l + 2}{2}\right] - l + 1 = p - c_0 - 2l + 2 = e$$

se tiene), p debe dividir $r - w$ para $1 \leq w \leq t - 1$, o p debe dividir $2r - w - 1$ para $t - e + 1 \leq w \leq t + e - 3$. Pero $0 < l \leq r - w \leq p - c_0 - l < p$ para $1 \leq w \leq t - 1 = p - c_0 - 2l + 1$, y, si $t - e + 1 \leq w \leq t + e - 3$, $1 \leq w \leq 2p - 2c_0 - 4l + 1$, luego

$$0 < 2l \leq 2r - w - 1 = 2p - 2c_0 - 2l + 1 - w \leq 2p - 2c_0 - 2l$$

y, por hipótesis, $2c_0 + 2l \geq p + 3$, de donde $2p - 2c_0 - 2l \leq p - 3 < p$. Por consiguiente, el determinante es múltiplo de p. Concluimos que $x_l = 0$. □

Lema 4.1.2. *Supongamos que* $(p + 3)/2 \leq 2l + c_0 \leq p \leq 2c_0 + 2l - 3$ *y* $c_0 + 1 \leq 2l$. *Entonces* $2c \geq m - p - 2l + c_0 - 1$.

Demostración. Supongamos que $2l + p - c_0 + 1 \leq m - 2c - 1$. Considerando pares de valores adyacentes no nulos en las filas $2l + 1$ y $2l + 2$ del triángulo $\mathcal{T}_G$, obtenemos que $z_i = z_{2l+1+c_0}$ para $1 \leq i \leq l - 1$ y que $z_i = z_{2l+1+c_0}$ para $l + 1 \leq i \leq 2l$.

Supongamos que $c_0 + 1 \leq 2l$. Consideremos

$$\alpha_{1,2l-c_0} = \sum_{u=0}^{2l-1} (-1)^u \binom{2l-1}{u} z_{1+u}.$$

Sabemos que $z_{1+u} = z_{2l+c_0+1}$ cuando $u \neq l - 1$, luego tenemos que

$$\alpha_{1,2l-c_0} = (-1)^{l-1} 2l - 1l - 1 z_l - (-1)^{l-1} 2l - 1l - 1 z_{2l+1+c_0} = 0,$$

teniendo en cuenta que $\sum_{u=0}^{2l-1} (-1)^u \binom{2l-1}{u} = 0$. Por tanto, obtenemos que $z_l = z_{2l+1+c_0}$, y tenemos la condición $z_1 = z_j$ para $1 \leq j \leq 2l$.

Un argumento inductivo nos muestra que $z_1 = z_j$ para $1 \leq j \leq c_0 + 2l$. Efectivamente, si tenemos que $z_1 = z_j$ para $1 \leq j \leq k$, con $2l \leq k < c_0 + 2l$, podemos considerar

$$\alpha_{1,k+1-c_0} = \sum_{u=0}^{k} (-1)^u \binom{k}{u} z_{1+u},$$

y teniendo en cuenta que $\sum_{u=0}^{k-1}(-1)^u k-1u=0$, obtenemos que

$$\alpha_{1,k+1-c_0}=(-1)^k z_{k+1}-(-1)^k z_1=0,$$

de donde $z_1=z_{k+1}$.

Recordando que $(p+3)/2\leq 2l+c_0$, deducimos que $p\leq 4l+2c_0-3$, so $p-c_0-2l+3\leq 2l+c_0$. Por el lema anterior, se llega a una contradicción. □

Lema 4.1.3. *Supongamos que $c_0\geq 2l$ y $2l+c_0\leq p\leq 3l+c_0-4$. Entonces $2c\geq m-(p+2l-c_0+1)$.*

Demostración. Observemos que, si $c_0\geq 2l$, $3l+c_0-4\leq l+2c_0-4\leq 2c_0+2l-3$, ya que $-4\leq l-3$.

Supongamos que $p+2l-c_0+1\leq m-2c-1$. Como $p\leq 3l+c_0-4$, $p-2l-c_0+3\leq l-1$. Además, $z_1=z_j$ para $j\leq l-1$. Por consiguiente, $z_j=z_1$ para $1\leq j\leq p-2l-c_0+3$. Como $2l+c_0\leq p\leq 3l+c_0-4$, llegamos a la contradicción $x_l=0$. □

Lema 4.1.4. *Supongamos que $c_0\geq 2l$ y $l+c_0+2\leq p\leq 2l+c_0-1$. Entonces $2c\geq m-(p+2l-c_0+1)$.*

Demostración. Supongamos que $p+2l-c_0+1\leq m-2c-1$. Como $p\geq l+c_0+2$, $2l+1-p+c_0\leq l-1$. En consecuencia, $z_j=z_1$ para $1\leq j\leq 2l+1-p+c_0$.

Por otra parte, $z_1=\alpha_{1,p-c_0}=0$, ya que $p-c_0+1\leq 2l$, y $z_{2l+1-p+c_0}=\alpha_{2l+1-p+c_0,p-c_0}=0$, una contradicción, ya que $(2l+1-p+c_0)+(p-c_0)=2l+1$. □

Obsérvese que hemos probado que, si $c_0\geq 2l$ y $l+c_0+2\leq p\leq 3l+c_0-4$, entonces $2c\geq m-(p+2l-c_0+1)$.

Lema 4.1.5. *Supongamos que $l\geq 3$, $4l\leq p+5$ y $c_0=p-3l+3$. Entonces $2c\geq m-(p+2l-c_0+1)$.*

Demostración. Supongamos que $l\geq 3$, $4l\leq p+5$ y $c_0=p-3l+3$, y $p+2l-c_0+1\leq m-2c-1$. En primer lugar, tenemos que $z_1=z_j$ para $j\leq l-1$.

De este modo, para $1 \leq i \leq l-2$, tenemos que $\alpha_{i,p-c_0+1} = \alpha_{i,3l-2} = 0$. Pero

$$\begin{aligned}\alpha_{i,3l-2} &= \sum_{k=i}^{[(3l-3+i)/2]} (-1)^{k-i} \binom{3l-3-k}{k-i} x_k \\ &= \sum_{k=i}^{[(3l-3+i)/2]} (-1)^{k-i} \binom{3l-3-k3l-3-2k+i}{x}_k \\ &= \sum_{s=1}^{[(l-1+i)/2]} (-1)^{s+l-1-i} \binom{2l-2-s}{l-1-2k+i} x_{s+l-1},\end{aligned}$$

y el número de incógnitas que aparecen aquí es $[(l-1+l-2)/2] = l-2$, luego, si llamamos $r = 2l-2$, $t = l-1$, $e = l-2$, sabemos que el determinante se anula si, y sólo si, existe un w con $1 \leq w \leq t-1 = l-2$ tal que p divide $(r-w) = 2l-2-w$ o existe w con $t-e+1 \leq w \leq t+e-3$, esto es, $2 \leq w \leq 2l-6$, tal que p divide $(2r-w-1) = 4l-5-w$. Pero, en el primer caso, tenemos que $0 < l \leq 2l-2-w \leq 2l-3 < p$, y, en el segundo caso, $0 < 2l+1 \leq 4l-5-w \leq 4l-7 \leq p-2 < p$ por la hipótesis $4l \leq p+5$. En consecuencia, tenemos que el determinante no es cero y $x_l = 0$, una contradicción. □

4.2. Las conjeturas

Utilizando los algoritmos vistos en el capítulo anterior, confeccionamos unas tablas de doble entrada para cada primo p en las que mostramos las cotas para el grado de conmutatividad en la forma $2c \geq m - g(c_0, l, p)$, de modo que la entrada que el índice de filas es c_0, el índice de columnas es l y la entrada en la posición correspondiente a la fila c_0 y columna l es el valor que estimamos para $g(c_0, l, p)$.

$p = 5$	$l = 1$
$c_0 = 0$	6
$c_0 = 1$	6
$c_0 = 2$	6
$c_0 = 3$	6

Cuadro 4.1: Cotas para $p = 5$ (Blackburn, [2])

$p = 7$	$l = 1$	$l = 2$
$c_0 = 0$	8	6
$c_0 = 1$	8	6
$c_0 = 2$	8	9
$c_0 = 3$	8	9
$c_0 = 4$	8	8
$c_0 = 5$	8	6

Cuadro 4.2: Cotas para $p = 7$ (Shepherd, [22])

$p = 11$	$l = 1$	$l = 2$	$l = 3$	$l = 4$
$c_0 = 0$	12	6	8	10
$c_0 = 1$	11	9	9	10
$c_0 = 2$	11	8	9	17
$c_0 = 3$	9	9	14	17
$c_0 = 4$	10	11	14	15
$c_0 = 5$	12	11	13	14
$c_0 = 6$	10	11	12	13
$c_0 = 7$	11	9	12	12
$c_0 = 8$	12	12	9	11
$c_0 = 9$	12	6	8	10

Cuadro 4.3: Cotas para $p = 11$

$p = 13$	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$
$c_0 = 0$	14	6	8	10	12
$c_0 = 1$	13	9	9	11	12
$c_0 = 2$	12	10	10	11	21
$c_0 = 3$	12	10	10	18	21
$c_0 = 4$	11	10	15	18	19
$c_0 = 5$	11	12	15	17	18
$c_0 = 6$	14	12	14	15	17
$c_0 = 7$	12	12	13	15	16
$c_0 = 8$	11	11	12	14	15
$c_0 = 9$	12	9	14	12	14
$c_0 = 10$	14	14	9	11	13
$c_0 = 11$	14	6	8	10	12

Cuadro 4.4: Cotas para $p = 13$

$p = 17$	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$	$l = 7$
$c_0 = 0$	18	6	8	10	12	14	16
$c_0 = 1$	17	9	9	11	13	15	16
$c_0 = 2$	18	10	10	12	14	15	29
$c_0 = 3$	17	11	11	13	14	26	29
$c_0 = 4$	16	12	12	13	23	26	27
$c_0 = 5$	16	16	12	20	23	25	26
$c_0 = 6$	15	12	17	20	22	23	25
$c_0 = 7$	14	14	17	19	21	23	24
$c_0 = 8$	18	14	16	18	20	21	23
$c_0 = 9$	16	13	15	17	19	20	22
$c_0 = 10$	14	13	15	16	18	19	21
$c_0 = 11$	14	12	13	15	18	18	20
$c_0 = 12$	14	11	12	18	15	17	19
$c_0 = 13$	16	9	18	12	14	16	18
$c_0 = 14$	18	18	9	11	13	15	17
$c_0 = 15$	18	6	8	10	12	14	16

Cuadro 4.5: Cotas para $p = 17$

$p = 19$	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$	$l = 7$	$l = 8$
$c_0 = 0$	20	6	8	10	12	14	16	18
$c_0 = 1$	21	9	9	11	13	15	17	18
$c_0 = 2$	20	10	10	12	14	16	17	33
$c_0 = 3$	20	11	11	13	15	16	30	33
$c_0 = 4$	18	12	12	14	15	27	30	31
$c_0 = 5$	17	14	13	14	24	27	29	30
$c_0 = 6$	17	17	13	21	24	26	27	29
$c_0 = 7$	17	15	18	21	23	25	27	28
$c_0 = 8$	16	15	18	20	22	24	25	27
$c_0 = 9$	20	15	17	19	21	23	24	26
$c_0 = 10$	18	14	16	18	20	21	23	25
$c_0 = 11$	16	14	15	17	19	21	22	24
$c_0 = 12$	15	13	15	16	18	20	21	23
$c_0 = 13$	17	12	13	15	20	18	20	22
$c_0 = 14$	16	11	12	20	15	17	19	21
$c_0 = 15$	18	9	20	12	14	16	18	20
$c_0 = 16$	20	20	9	11	13	15	17	19
$c_0 = 17$	20	6	8	10	12	14	16	18

Cuadro 4.6: Cotas para $p = 19$

$p = 23$	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$	$l = 7$	$l = 8$	$l = 9$	$l = 10$
$c_0 = 0$	24	6	8	10	12	14	16	18	20	22
$c_0 = 1$	25	9	9	11	13	15	17	19	21	22
$c_0 = 2$	24	10	10	12	14	16	18	20	21	41
$c_0 = 3$	24	11	11	13	15	17	19	20	38	41
$c_0 = 4$	23	14	12	14	16	18	19	35	38	39
$c_0 = 5$	23	15	13	15	17	18	32	35	37	38
$c_0 = 6$	22	17	14	16	17	29	32	34	35	37
$c_0 = 7$	20	18	15	16	26	29	31	33	35	36
$c_0 = 8$	20	19	15	23	26	28	30	32	33	35
$c_0 = 9$	19	18	20	23	25	27	29	31	32	34
$c_0 = 10$	19	17	20	22	24	26	28	29	31	33
$c_0 = 11$	24	17	19	21	23	25	27	29	30	32
$c_0 = 12$	22	16	18	20	22	24	26	27	29	31
$c_0 = 13$	20	15	18	19	21	23	25	26	28	30
$c_0 = 14$	18	15	17	18	20	22	24	25	27	29
$c_0 = 15$	18	14	15	17	19	21	24	24	26	28
$c_0 = 16$	20	13	15	16	18	24	21	23	25	27
$c_0 = 17$	18	12	13	15	24	18	20	22	24	26
$c_0 = 18$	20	11	12	24	15	17	19	21	23	25
$c_0 = 19$	22	9	24	12	14	16	18	20	22	24
$c_0 = 20$	24	24	9	11	13	15	17	19	21	23
$c_0 = 21$	24	6	8	10	12	14	16	18	20	22

Cuadro 4.7: Cotas para $p = 23$

$p = 29$	1	2	3	4	5	6	7	8	9	10	11	12	13
$c_0 = 0$	30	6	8	10	12	14	16	18	20	22	24	26	28
$c_0 = 1$	31	9	9	11	13	15	17	19	21	23	25	27	28
$c_0 = 2$	30	10	10	12	14	16	18	20	22	24	26	27	53
$c_0 = 3$	30	11	11	13	15	17	19	21	23	25	26	50	53
$c_0 = 4$	29	12	12	14	16	18	20	22	24	25	47	50	51
$c_0 = 5$	28	15	13	15	17	19	21	23	24	44	47	49	50
$c_0 = 6$	28	17	14	16	18	20	22	23	41	44	46	47	49
$c_0 = 7$	27	23	15	17	19	21	22	38	41	43	45	47	48
$c_0 = 8$	26	22	16	18	20	21	35	38	40	42	44	45	47
$c_0 = 9$	25	21	17	19	20	32	35	37	39	41	43	44	46
$c_0 = 10$	24	23	19	19	29	32	34	36	38	40	41	43	45
$c_0 = 11$	23	23	19	26	29	31	33	35	37	39	41	42	44
$c_0 = 12$	23	21	23	26	28	30	32	34	36	38	39	41	43
$c_0 = 13$	22	21	23	25	27	29	31	33	35	37	38	40	42
$c_0 = 14$	30	21	22	24	26	28	30	32	34	35	37	39	41
$c_0 = 15$	28	20	21	23	25	27	29	31	33	35	36	38	40
$c_0 = 16$	26	19	20	22	24	26	28	30	32	33	35	37	39
$c_0 = 17$	24	18	20	21	23	25	27	29	31	32	34	36	38
$c_0 = 18$	22	17	19	21	22	24	26	28	30	31	33	35	37
$c_0 = 19$	22	16	18	19	21	23	25	27	30	30	32	34	36
$c_0 = 20$	21	15	17	18	20	22	24	30	27	29	31	33	35
$c_0 = 21$	23	14	15	17	19	21	30	24	26	28	30	32	34
$c_0 = 22$	22	13	15	16	18	30	21	23	25	27	29	31	33
$c_0 = 23$	24	12	13	15	30	18	20	22	24	26	28	30	32
$c_0 = 24$	26	11	12	30	15	17	19	21	23	25	27	29	31
$c_0 = 25$	28	9	30	12	14	16	18	20	22	24	26	28	30
$c_0 = 26$	30	30	9	11	13	15	17	19	21	23	25	27	29
$c_0 = 27$	30	6	8	10	12	14	16	18	20	22	24	26	28

Cuadro 4.8: Cotas para $p = 29$

$p = 31$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$c_0 = 0$	32	6	8	10	12	14	16	18	20	22	24	26	28	30
$c_0 = 1$	33	9	9	11	13	15	17	19	21	23	25	27	29	30
$c_0 = 2$	32	10	10	12	14	16	18	20	22	24	26	28	29	57
$c_0 = 3$	32	11	11	13	15	17	19	21	23	25	27	28	54	57
$c_0 = 4$	32	12	12	14	16	18	20	22	24	26	27	51	54	55
$c_0 = 5$	31	15	13	15	17	19	21	23	25	26	48	51	53	54
$c_0 = 6$	30	17	14	16	18	20	22	24	25	45	48	50	51	53
$c_0 = 7$	29	18	15	17	19	21	23	24	42	45	47	49	51	52
$c_0 = 8$	28	24	16	18	20	22	23	39	42	44	46	48	49	51
$c_0 = 9$	27	23	19	19	21	22	36	39	41	43	45	47	48	50
$c_0 = 10$	26	25	18	20	21	33	36	38	40	42	44	45	47	49
$c_0 = 11$	25	24	21	20	30	33	35	37	39	41	43	45	46	48
$c_0 = 12$	24	24	20	27	30	32	34	36	38	40	42	43	45	47
$c_0 = 13$	24	23	24	27	29	31	33	35	37	39	41	42	44	46
$c_0 = 14$	25	22	24	26	28	30	32	34	36	38	39	41	43	45
$c_0 = 15$	32	22	23	25	27	29	31	33	35	37	39	40	42	44
$c_0 = 16$	30	21	22	24	26	28	30	32	34	36	37	39	41	43
$c_0 = 17$	28	20	21	23	25	27	29	31	33	35	36	38	40	42
$c_0 = 18$	26	19	21	23	24	26	28	30	32	33	35	37	39	41
$c_0 = 19$	24	18	20	21	23	25	27	29	31	33	34	36	38	40
$c_0 = 20$	23	17	19	21	22	24	26	28	30	32	33	35	37	39
$c_0 = 21$	22	16	18	19	21	23	25	27	32	30	32	34	36	38
$c_0 = 22$	26	15	17	18	20	22	24	32	27	29	31	33	35	37
$c_0 = 23$	23	14	15	17	19	21	32	24	26	28	30	32	34	36
$c_0 = 24$	24	13	15	16	18	32	21	23	25	27	29	31	33	35
$c_0 = 25$	26	12	13	15	32	18	20	22	24	26	28	30	32	34
$c_0 = 26$	28	11	12	32	15	17	19	21	23	25	27	29	31	33
$c_0 = 27$	30	9	32	12	14	16	18	20	22	24	26	28	30	32
$c_0 = 28$	32	32	9	11	13	15	17	19	21	23	25	27	29	31
$c_0 = 29$	32	6	8	10	12	14	16	18	20	22	24	26	28	30

Cuadro 4.9: Cotas para $p = 31$

$p = 37$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$c_0 = 0$	38	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36
$c_0 = 1$	39	9	9	11	13	15	17	19	21	23	25	27	29	31	33	35	36
$c_0 = 2$	38	10	10	12	14	16	18	20	22	24	26	28	30	32	34	35	69
$c_0 = 3$	38	11	11	13	15	17	19	21	23	25	27	29	31	33	34	66	69
$c_0 = 4$	38	12	12	14	16	18	20	22	24	26	28	30	32	33	63	66	67
$c_0 = 5$	37	15	13	15	17	19	21	23	25	27	29	31	32	60	63	65	66
$c_0 = 6$	36	17	14	16	18	20	22	24	26	28	30	31	57	60	62	63	65
$c_0 = 7$	35	18	15	17	19	21	23	25	27	29	30	54	57	59	61	63	64
$c_0 = 8$	35	28	16	18	20	22	24	26	28	29	51	54	56	58	60	61	63
$c_0 = 9$	33	31	24	19	21	23	25	27	28	48	51	53	55	57	59	60	62
$c_0 = 10$	32	28	18	20	22	24	26	27	45	48	50	52	54	56	57	59	61
$c_0 = 11$	32	27	28	21	23	25	26	42	45	47	49	51	53	55	57	58	60
$c_0 = 12$	31	29	31	22	24	25	39	42	44	46	48	50	52	54	55	57	59
$c_0 = 13$	30	29	30	25	24	36	39	41	43	45	47	49	51	53	54	56	58
$c_0 = 14$	29	28	24	23	33	36	38	40	42	44	46	48	50	51	53	55	57
$c_0 = 15$	28	29	23	30	33	35	37	39	41	43	45	47	49	51	52	54	56
$c_0 = 16$	29	28	27	30	32	34	36	38	40	42	44	46	48	49	51	53	55
$c_0 = 17$	29	26	27	29	31	33	35	37	39	41	43	45	47	48	50	52	54
$c_0 = 18$	38	26	26	28	30	32	34	36	38	40	42	44	45	47	49	51	53
$c_0 = 19$	36	25	25	27	29	31	33	35	37	39	41	43	45	46	48	50	52
$c_0 = 20$	34	24	24	26	28	30	32	34	36	38	40	42	43	45	47	49	51
$c_0 = 21$	32	23	23	25	27	29	31	33	35	37	39	41	42	44	46	48	50
$c_0 = 22$	30	22	23	25	27	28	30	32	34	36	38	39	41	43	45	47	49
$c_0 = 23$	28	21	22	24	25	27	29	31	33	35	37	39	40	42	44	46	48
$c_0 = 24$	27	19	21	23	24	26	28	30	32	34	36	38	39	41	43	45	47
$c_0 = 25$	26	18	20	21	23	25	27	29	31	33	38	36	38	40	42	44	46
$c_0 = 26$	28	17	19	21	22	24	26	28	30	38	33	35	37	39	41	43	45
$c_0 = 27$	29	16	18	19	21	23	25	27	38	30	32	34	36	38	40	42	44
$c_0 = 28$	26	15	17	18	20	22	24	38	27	29	31	33	35	37	39	41	43
$c_0 = 29$	28	14	15	17	19	21	38	24	26	28	30	32	34	36	38	40	42
$c_0 = 30$	30	13	15	16	18	38	21	23	25	27	29	31	33	35	37	39	41
$c_0 = 31$	32	12	13	15	38	18	20	22	24	26	28	30	32	34	36	38	40
$c_0 = 32$	34	11	12	38	15	17	19	21	23	25	27	29	31	33	35	37	39
$c_0 = 33$	36	9	38	12	14	16	18	20	22	24	26	28	30	32	34	36	38
$c_0 = 34$	38	38	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37
$c_0 = 35$	38	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36

Cuadro 4.10: Cotas para $p = 37$

$p=41$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$c_0=0$	42	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
$c_0=1$	43	9	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	40
$c_0=2$	42	10	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	39	77
$c_0=3$	42	11	11	13	15	17	19	21	23	25	27	29	31	33	35	37	38	74	77
$c_0=4$	42	12	12	14	16	18	20	22	24	26	28	30	32	34	36	37	71	74	75
$c_0=5$	41	15	13	15	17	19	21	23	25	27	29	31	33	35	36	68	71	73	74
$c_0=6$	40	17	14	16	18	20	22	24	26	28	30	32	34	35	65	68	70	71	73
$c_0=7$	39	18	15	17	19	21	23	25	27	29	31	33	34	62	65	67	69	71	72
$c_0=8$	39	28	16	18	20	22	24	26	28	30	32	33	59	62	64	66	68	69	71
$c_0=9$	37	31	17	19	21	23	25	27	29	31	32	56	59	61	63	65	67	68	70
$c_0=10$	36	33	18	20	22	24	26	28	30	31	53	56	58	60	62	64	65	67	69
$c_0=11$	36	31	28	21	23	25	27	29	30	50	53	55	57	59	61	63	65	66	68
$c_0=12$	34	30	29	22	24	26	28	29	47	50	52	54	56	58	60	62	63	65	67
$c_0=13$	34	30	34	23	25	27	28	44	47	49	51	53	55	57	59	61	62	64	66
$c_0=14$	33	31	33	24	26	27	41	44	46	48	50	52	54	56	58	59	61	63	65
$c_0=15$	32	31	32	27	26	38	41	43	45	47	49	51	53	55	57	59	60	62	64
$c_0=16$	31	31	31	26	35	38	40	42	44	46	48	50	52	54	56	57	59	61	63
$c_0=17$	31	32	27	32	35	37	39	41	43	45	47	49	51	53	55	56	58	60	62
$c_0=18$	32	31	29	32	34	36	38	40	42	44	46	48	50	52	53	55	57	59	61
$c_0=19$	33	30	29	31	33	35	37	39	41	43	45	47	49	51	53	54	56	58	60
$c_0=20$	42	28	28	30	32	34	36	38	40	42	44	46	48	50	51	53	55	57	59
$c_0=21$	40	29	27	29	31	33	35	37	39	41	43	45	47	49	50	52	54	56	58
$c_0=22$	38	27	26	28	30	32	34	36	38	40	42	44	46	47	49	51	53	55	57
$c_0=23$	36	26	25	27	29	31	33	35	37	39	41	43	45	47	48	50	52	54	56
$c_0=24$	34	25	24	27	29	30	32	34	36	38	40	42	44	45	47	49	51	53	55
$c_0=25$	32	24	24	26	27	29	31	33	35	37	39	41	43	44	46	48	50	52	54
$c_0=26$	30	23	23	25	27	28	30	32	34	36	38	40	42	43	45	47	49	51	53
$c_0=27$	30	21	22	24	25	27	29	31	33	35	37	39	42	42	44	46	48	50	52
$c_0=28$	29	21	21	23	24	26	28	30	32	34	36	42	39	41	43	45	47	49	51
$c_0=29$	28	19	20	21	23	25	27	29	31	33	42	36	38	40	42	44	46	48	50
$c_0=30$	32	17	19	21	22	24	26	28	30	42	33	35	37	39	41	43	45	47	49
$c_0=31$	29	16	18	19	21	23	25	27	42	30	32	34	36	38	40	42	44	46	48
$c_0=32$	30	15	17	18	20	22	24	42	27	29	31	33	35	37	39	41	43	45	47
$c_0=33$	32	14	15	17	19	21	42	24	26	28	30	32	34	36	38	40	42	44	46
$c_0=34$	34	13	15	16	18	42	21	23	25	27	29	31	33	35	37	39	41	43	45
$c_0=35$	36	12	13	15	42	18	20	22	24	26	28	30	32	34	36	38	40	42	44
$c_0=36$	38	11	12	42	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43
$c_0=37$	40	9	42	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42
$c_0=38$	42	42	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41
$c_0=39$	42	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40

Cuadro 4.11: Cotas para $p=41$

Blackburn dio en [2] la cota $2c \geq m-6$ para $p=5$ para cualquier valor de c_0 (véase la tabla 4.1). Shepherd probó en [22, Theorem 1.27] la cota $2c \geq m-8$ para $c_0 \in \{0,1,4,5\}$ y $2c \geq m-9$ para $c_0 \in \{2,3\}$, como se puede observar en la tabla 4.2.

La observación de las tablas 4.1 a 4.12 nos lleva a realizar las siguientes conjeturas:

Conjetura A. *1. Si* $\frac{p+7}{6} \leq l = \frac{p+1}{2} - c_0$, *entonces* $2c \geq m - p - 2l + c_0$.

2. Si $p - c_0 \leq l \leq \frac{p-3}{2}$, *entonces* $2c \geq m - p - 2l + c_0$.

3. Si $c_0 \geq 2p - 4l - 2$ *y* $3l > p$, *o* $c_0 = 2p - 4l - 4$ *y* $3l > p$, *entonces* $2c \geq m - p - 2l + c_0$.

$p = 43$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$c_0 = 0$	44	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42
$c_0 = 1$	45	9	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	42
$c_0 = 2$	44	10	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	41	81
$c_0 = 3$	44	11	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	40	78	81
$c_0 = 4$	44	12	12	14	16	18	20	22	24	26	28	30	32	34	36	38	39	75	78	79
$c_0 = 5$	43	15	13	15	17	19	21	23	25	27	29	31	33	35	37	38	72	75	77	78
$c_0 = 6$	42	17	14	16	18	20	22	24	26	28	30	32	34	36	37	69	72	74	75	77
$c_0 = 7$	41	18	15	17	19	21	23	25	27	29	31	33	35	36	66	69	71	73	75	76
$c_0 = 8$	40	28	16	18	20	22	24	26	28	30	32	34	35	63	66	68	70	72	73	75
$c_0 = 9$	39	31	24	19	21	23	25	27	29	31	33	34	60	63	65	67	69	71	72	74
$c_0 = 10$	38	33	25	20	22	24	26	28	30	32	33	57	60	62	64	66	68	69	71	73
$c_0 = 11$	37	35	28	21	23	25	27	29	31	32	54	57	59	61	63	65	67	69	70	72
$c_0 = 12$	36	32	30	22	24	26	28	30	31	51	54	56	58	60	62	64	66	67	69	71
$c_0 = 13$	36	31	30	23	25	27	29	30	48	51	53	55	57	59	61	63	65	66	68	70
$c_0 = 14$	35	33	35	25	26	28	29	45	48	50	52	54	56	58	60	62	63	65	67	69
$c_0 = 15$	34	32	34	29	27	28	42	45	47	49	51	53	55	57	59	61	63	64	66	68
$c_0 = 16$	33	32	33	28	27	39	42	44	46	48	50	52	54	56	58	60	61	63	65	67
$c_0 = 17$	32	33	32	27	36	39	41	43	45	47	49	51	53	55	57	59	60	62	64	66
$c_0 = 18$	33	33	31	33	36	38	40	42	44	46	48	50	52	54	56	57	59	61	63	65
$c_0 = 19$	33	33	30	33	35	37	39	41	43	45	47	49	51	53	55	57	58	60	62	64
$c_0 = 20$	34	31	30	32	34	36	38	40	42	44	46	48	50	52	54	55	57	59	61	63
$c_0 = 21$	44	29	29	31	33	35	37	39	41	43	45	47	49	51	53	54	56	58	60	62
$c_0 = 22$	42	30	28	30	32	34	36	38	40	42	44	46	48	50	51	53	55	57	59	61
$c_0 = 23$	40	29	27	29	31	33	35	37	39	41	43	45	47	49	51	52	54	56	58	60
$c_0 = 24$	38	28	26	28	30	32	34	36	38	40	42	44	46	48	49	51	53	55	57	59
$c_0 = 25$	36	27	25	27	30	31	33	35	37	39	41	43	45	47	48	50	52	54	56	58
$c_0 = 26$	34	26	25	27	29	30	32	34	36	38	40	42	44	45	47	49	51	53	55	57
$c_0 = 27$	32	24	24	26	27	29	31	33	35	37	39	41	43	45	46	48	50	52	54	56
$c_0 = 28$	32	23	23	25	27	28	30	32	34	36	38	40	42	44	45	47	49	51	53	55
$c_0 = 29$	30	21	22	24	25	27	29	31	33	35	37	39	44	42	44	46	48	50	52	54
$c_0 = 30$	28	19	21	23	24	26	28	30	32	34	36	44	39	41	43	45	47	49	51	53
$c_0 = 31$	35	18	20	21	23	25	27	29	31	33	44	36	38	40	42	44	46	48	50	52
$c_0 = 32$	32	17	19	21	22	24	26	28	30	44	33	35	37	39	41	43	45	47	49	51
$c_0 = 33$	30	16	18	19	21	23	25	27	44	30	32	34	36	38	40	42	44	46	48	50
$c_0 = 34$	32	15	17	18	20	22	24	44	27	29	31	33	35	37	39	41	43	45	47	49
$c_0 = 35$	34	14	15	17	19	21	44	24	26	28	30	32	34	36	38	40	42	44	46	48
$c_0 = 36$	36	13	15	16	18	44	21	23	25	27	29	31	33	35	37	39	41	43	45	47
$c_0 = 37$	38	12	13	15	44	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46
$c_0 = 38$	40	11	12	44	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45
$c_0 = 39$	42	9	44	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44
$c_0 = 40$	44	44	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43
$c_0 = 41$	44	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42

Cuadro 4.12: Cotas para $p = 43$

Conjetura B. *1. Si* $p+6 \leq c_0+4l \leq 2p-5$, *entonces* $2c \geq m-p-2l+c_0-1$.

2. Si $p+4 = c_0+4l \leq 2p-5$, *entonces* $2c \geq m-p-2l+c_0-1$.

3. Si $p+6 \leq c_0+4l = 2p-3$, *entonces* $2c \geq m-p-2l+c_0-1$.

4. Si $l \geq 3$, $\frac{p+1}{2} \leq c_0+l-1$, $c_0+2 < \frac{2(p-1)}{3}$ *y* $c_0+4l \leq 2p-5$ *o* $c_0+4l = 2p-3$, *then* $2c \geq m-p-2l+c_0-1$.

5. Si $c_0+l = 3l = p-2$, *entonces* $2c \geq m-p-2l+c_0-1$.

Conjetura C. *Si* $l = p-c_0-1$ *y* $3l < p$, *entonces* $2c \geq m-p-1$.

Conjetura D. *1. Si* $2+\frac{c_0}{7} \leq l \leq \frac{p-5}{2}-c_0$, *entonces* $2c \geq m-2l-c_0-2$.

2. Si $3+\frac{c_0}{7} \leq l = \frac{p-3}{2}-c_0$, *entonces* $2c \geq m-2l-c_0-2$.

Conjetura E. *Si* $\frac{p+5}{6} \leq l = \frac{p-1}{2}-c_0$, *entonces* $2c \geq m-2l-c_0-1$.

Conjetura F. *1. Si* $3 \leq l$, $c_0+4l = p+5$ *o* $c_0+4l \leq p+3$, *y* $c_0+3 \geq \frac{2}{3}p$, *entonces* $2c \geq m-p-2l+c_0-2$.

2. Si $l = 1$, $\frac{p-1}{2} \leq c_0 < \frac{2}{3}(p-1)$, *entonces* $2c \geq m-2p+2c_0$.

Conjetura G. *Si* $l = 1$, $p \geq 13$ *y* $p-2-\left[\frac{p-1}{6}\right] \leq c_0 \leq p-2$, *entonces* $2c \geq m-p-1+2(p-c_0-3)$.

Para las casillas de la tabla que no corresponden a las regiones cubiertas por las conjeturas previas, observamos que los valores asociados son menores que $3p/4$.

El siguiente Lema será utilizado en las Secciones 4.3 y 4.5.

Lema 4.2.1. *Si* $3l \leq m-2c-1$ *y* $l \geq 2$, *entonces* $\alpha_{1,2l+c_0+k} = 0$ *para* $1 \leq k \leq l-1$.

Demostración. Tenemos que $3l \leq m-2c-1$, luego podemos aplicar la identidad de Jacobi para las ternas $(l-k, l, l+1)$, siendo $k \in \{1, \ldots, l-1\}$. Entonces obtenemos que

$$0 = x_l \alpha_{2l+1+c_0, l-k},$$

$p = 17$	1	2	3	4	5	6	7
0		6	8	10	12	14	16
1			9	11	13	15	16
2			10	12	14	15	29
3			11	13	14	26	29
4				13	23	26	27
5			12	20	23	25	26
6				20	22	23	25
7			17	19	21	23	24
8	18		17	18	20	21	23
9	16		15	17	19	20	22
10			15	16	18	19	21
11			13	15	18	18	20
12			12	18	15	17	19
13	16	9	18	12	14	16	18
14	18	18	9	11	13	15	17
15	18	6	8	10	12	14	16

Cuadro 4.13: Cotas conjeturadas para $p = 17$

a

$p = 19$	1	2	3	4	5	6	7	8
0		6	8	10	12	14	16	18
1			9	11	13	15	17	18
2			10	12	14	16	17	33
3			11	13	15	16	30	33
4			12	14	15	27	30	31
5				14	24	27	29	30
6				21	24	26	27	29
7				21	23	25	27	28
8			18	20	22	24	25	27
9	20		17	19	21	23	24	26
10	18		17	18	20	21	23	25
11	16		15	17	19	21	22	24
12			15	16	18	20	21	23
13			13	15	20	18	20	22
14	16		12	20	15	17	19	21
15	18	9	20	12	14	16	18	20
16	20	20	9	11	13	15	17	19
17	20	6	8	10	12	14	16	18

Cuadro 4.14: Cotas conjeturadas para $p = 19$

$p = 23$	1	2	3	4	5	6	7	8	9	10
0		6	8	10	12	14	16	18	20	22
1			9	11	13	15	17	19	21	22
2			10	12	14	16	18	20	21	41
3			11	13	15	17	19	20	38	41
4			12	14	16	18	19	35	38	39
5			13	15	17	18	32	35	37	38
6			14	16	17	29	32	34	35	37
7				16	26	29	31	33	35	36
8					26	28	30	32	33	35
9				23	25	27	29	31	32	34
10			20	22	24	26	28	29	31	33
11	24		19	21	23	25	27	29	30	32
12	22		19	21	22	24	26	27	29	31
13	20		18	19	21	23	25	26	28	30
14			17	18	20	22	24	25	27	29
15			15	17	19	21	24	24	26	28
16			15	16	18	24	21	23	25	27
17			13	15	24	18	20	22	24	26
18	20		12	24	15	17	19	21	23	25
19	22	9	24	12	14	16	18	20	22	24
20	24	24	9	11	13	15	17	19	21	23
21	24	6	8	10	12	14	16	18	20	22

Cuadro 4.15: Cotas conjeturadas para $p = 23$

$p = 29$	1	2	3	4	5	6	7	8	9	10	11	12	13
0		6	8	10	12	14	16	18	20	22	24	26	28
1			9	11	13	15	17	19	21	23	25	27	28
2			10	12	14	16	18	20	22	24	26	27	53
3			11	13	15	17	19	21	23	25	26	50	53
4			12	14	16	18	20	22	24	25	47	50	51
5			13	15	17	19	21	23	24	44	47	49	50
6			14	16	18	20	22	23	41	44	46	47	49
7			15	17	19	21	22	38	41	43	45	47	48
8				18	20	21	35	38	40	42	44	45	47
9					20	32	35	37	39	41	43	44	46
10						32	34	36	38	40	41	43	45
11					29	31	33	35	37	39	41	42	44
12				26	28	30	32	34	36	38	39	41	43
13			23	25	27	29	31	33	35	37	38	40	42
14	30		22	24	26	28	30	32	34	35	37	39	41
15	28		21	23	25	27	29	31	33	35	36	38	40
16	26		21	23	24	26	28	30	32	33	35	37	39
17	24		20	21	23	25	27	29	31	32	34	36	38
18			19	21	22	24	26	28	30	31	33	35	37
19			18	19	21	23	25	27	30	30	32	34	36
20			17	18	20	22	24	30	27	29	31	33	35
21			15	17	19	21	30	24	26	28	30	32	34
22			15	16	18	30	21	23	25	27	29	31	33
23	24		13	15	30	18	20	22	24	26	28	30	32
24	26		12	30	15	17	19	21	23	25	27	29	31
25	28	9	30	12	14	16	18	20	22	24	26	28	30
26	30	30	9	11	13	15	17	19	21	23	25	27	29
27	30	6	8	10	12	14	16	18	20	22	24	26	28

Cuadro 4.16: Cotas conjeturadas para $p = 29$

$p = 31$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0		6	8	10	12	14	16	18	20	22	24	26	28	30
1			9	11	13	15	17	19	21	23	25	27	29	30
2			10	12	14	16	18	20	22	24	26	28	29	57
3			11	13	15	17	19	21	23	25	27	28	54	57
4			12	14	16	18	20	22	24	26	27	51	54	55
5			13	15	17	19	21	23	25	26	48	51	53	54
6			14	16	18	20	22	24	25	45	48	50	51	53
7			15	17	19	21	23	24	42	45	47	49	51	52
8				18	20	22	23	39	42	44	46	48	49	51
9				19	21	22	36	39	41	43	45	47	48	50
10						33	36	38	40	42	44	45	47	49
11						33	35	37	39	41	43	45	46	48
12					30	32	34	36	38	40	42	43	45	47
13				27	29	31	33	35	37	39	41	42	44	46
14			24	26	28	30	32	34	36	38	39	41	43	45
15	32		23	25	27	29	31	33	35	37	39	40	42	44
16	30		22	24	26	28	30	32	34	36	37	39	41	43
17	28		21	23	25	27	29	31	33	35	36	38	40	42
18	26		21	23	24	26	28	30	32	33	35	37	39	41
19	24		20	21	23	25	27	29	31	33	34	36	38	40
20			19	21	22	24	26	28	30	32	33	35	37	39
21			18	19	21	23	25	27	32	30	32	34	36	38
22			17	18	20	22	24	32	27	29	31	33	35	37
23			15	17	19	21	32	24	26	28	30	32	34	36
24	24		15	16	18	32	21	23	25	27	29	31	33	35
25	26		13	15	32	18	20	22	24	26	28	30	32	34
26	28		12	32	15	17	19	21	23	25	27	29	31	33
27	30	9	32	12	14	16	18	20	22	24	26	28	30	32
28	32	32	9	11	13	15	17	19	21	23	25	27	29	31
29	32	6	8	10	12	14	16	18	20	22	24	26	28	30

Cuadro 4.17: Cotas conjeturadas para $p = 31$

$p = 37$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0		6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36
1			9	11	13	15	17	19	21	23	25	27	29	31	33	35	36
2			10	12	14	16	18	20	22	24	26	28	30	32	34	35	69
3			11	13	15	17	19	21	23	25	27	29	31	33	34	66	69
4			12	14	16	18	20	22	24	26	28	30	32	33	63	66	67
5			13	15	17	19	21	23	25	27	29	31	32	60	63	65	66
6			14	16	18	20	22	24	26	28	30	31	57	60	62	63	65
7			15	17	19	21	23	25	27	29	30	54	57	59	61	63	64
8				18	20	22	24	26	28	29	51	54	56	58	60	61	63
9				19	21	23	25	27	28	48	51	53	55	57	59	60	62
10				20	22	24	26	27	45	48	50	52	54	56	57	59	61
11				21	23	25	26	42	45	47	49	51	53	55	57	58	60
12				22	24		39	42	44	46	48	50	52	54	55	57	59
13							39	41	43	45	47	49	51	53	54	56	58
14						36	38	40	42	44	46	48	50	51	53	55	57
15					33	35	37	39	41	43	45	47	49	51	52	54	56
16				30	32	34	36	38	40	42	44	46	48	49	51	53	55
17			27	29	31	33	35	37	39	41	43	45	47	48	50	52	54
18	38		26	28	30	32	34	36	38	40	42	44	45	47	49	51	53
19	36		25	27	29	31	33	35	37	39	41	43	45	46	48	50	52
20	34		24	26	28	30	32	34	36	38	40	42	43	45	47	49	51
21	32		23	25	27	29	31	33	35	37	39	41	42	44	46	48	50
22	30		23	25	27	28	30	32	34	36	38	39	41	43	45	47	49
23	28		22	24	25	27	29	31	33	35	37	39	40	42	44	46	48
24			21	23	24	26	28	30	32	34	36	38	39	41	43	45	47
25			20	21	23	25	27	29	31	33	38	36	38	40	42	44	46
26			19	21	22	24	26	28	30	38	33	35	37	39	41	43	45
27			18	19	21	23	25	27	38	30	32	34	36	38	40	42	44
28			17	18	20	22	24	38	27	29	31	33	35	37	39	41	43
29	28		15	17	19	21	38	24	26	28	30	32	34	36	38	40	42
30	30		15	16	18	38	21	23	25	27	29	31	33	35	37	39	41
31	32		13	15	38	18	20	22	24	26	28	30	32	34	36	38	40
32	34		12	38	15	17	19	21	23	25	27	29	31	33	35	37	39
33	36	9	38	12	14	16	18	20	22	24	26	28	30	32	34	36	38
34	38	38	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37
35	38	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36

Cuadro 4.18: Cotas conjeturadas para $p = 37$

$p = 41$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0		6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
1			9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	40
2			10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	39	77
3			11	13	15	17	19	21	23	25	27	29	31	33	35	37	38	74	77
4			12	14	16	18	20	22	24	26	28	30	32	34	36	37	71	74	75
5			13	15	17	19	21	23	25	27	29	31	33	35	36	68	71	73	74
6			14	16	18	20	22	24	26	28	30	32	34	35	65	68	70	71	73
7			15	17	19	21	23	25	27	29	31	33	34	62	65	67	69	71	72
8				18	20	22	24	26	28	30	32	33	59	62	64	66	68	69	71
9				19	21	23	25	27	29	31	32	56	59	61	63	65	67	68	70
10				20	22	24	26	28	30	31	53	56	58	60	62	64	65	67	69
11				21	23	25	27	29	30	50	53	55	57	59	61	63	65	66	68
12				22	24	26	28	29	47	50	52	54	56	58	60	62	63	65	67
13				23	25	27	28	44	47	49	51	53	55	57	59	61	62	64	66
14				24	26			44	46	48	50	52	54	56	58	59	61	63	65
15							41	43	45	47	49	51	53	55	57	59	60	62	64
16						38	40	42	44	46	48	50	52	54	56	57	59	61	63
17					35	37	39	41	43	45	47	49	51	53	55	56	58	60	62
18				32	34	36	38	40	42	44	46	48	50	52	53	55	57	59	61
19			29	31	33	35	37	39	41	43	45	47	49	51	53	54	56	58	60
20	42		28	30	32	34	36	38	40	42	44	46	48	50	51	53	55	57	59
21	40		27	29	31	33	35	37	39	41	43	45	47	49	50	52	54	56	58
22	38		26	28	30	32	34	36	38	40	42	44	46	47	49	51	53	55	57
23	36		25	27	29	31	33	35	37	39	41	43	45	47	48	50	52	54	56
24	34		25	27	29	30	32	34	36	38	40	42	44	45	47	49	51	53	55
25	32		24	26	27	29	31	33	35	37	39	41	43	44	46	48	50	52	54
26			23	25	27	28	30	32	34	36	38	40	42	43	45	47	49	51	53
27			22	24	25	27	29	31	33	35	37	39	42	42	44	46	48	50	52
28			21	23	24	26	28	30	32	34	36	42	39	41	43	45	47	49	51
29			20	21	23	25	27	29	31	33	42	36	38	40	42	44	46	48	50
30			19	21	22	24	26	28	30	42	33	35	37	39	41	43	45	47	49
31			18	19	21	23	25	27	42	30	32	34	36	38	40	42	44	46	48
32			17	18	20	22	24	42	27	29	31	33	35	37	39	41	43	45	47
33	32		15	17	19	21	42	24	26	28	30	32	34	36	38	40	42	44	46
34	34		15	16	18	42	21	23	25	27	29	31	33	35	37	39	41	43	45
35	36		13	15	42	18	20	22	24	26	28	30	32	34	36	38	40	42	44
36	38		12	42	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43
37	40	9	42	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42
38	42	42	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41
39	42	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40

Cuadro 4.19: Cotas conjeturadas para $p = 41$

de donde $\alpha_{2l+1+c_0,l-k} = 0$. Por tanto, tenemos la siguiente diagonal de ceros:

$$\alpha_{l-k,2l+1+c_0} = 0, \qquad k \in \{1, \ldots, l-1\},$$

pero usando la condición $\alpha_{i,j} = \alpha_{i+1,j} + \alpha_{i,j+1}$, deducimos que

$$\alpha_{1,2l+c_0+k} = 0, \qquad k \in \{1, \ldots, l-1\}.$$

□

4.3. Cotas de la forma $2c \geq m - p - 2l + c_0$.

Los siguientes resultados son interesantes para probar algunas cotas cuando c_0 y l son "grandes" en relación con p, ya que podemos evitar en algunos casos el cálculo de algunas ecuaciones de Jacobi.

Lema 4.3.1. *Supongamos que $p \leq 2l + c_0 + 1$. Si $i + j + k \leq p + 2l - c_0 - 1$, entonces $\alpha_{i,j}\alpha_{i+j+c_0,k} = 0$.*

Demostración. Supongamos que $i + j + k \leq p + 2l - c_0 - 1$.
Si $i + j \leq 2l$, entonces $\alpha_{i,j} = 0$.
Si $i + j \geq 2l + 1$, debemos probar que $\alpha_{i+j+c_0,k} = 0$. En primer lugar, tenemos que

$$i + j + c_0 \geq 2l + c_0 + 1 \geq p,$$

luego podemos considerar

$$i + j + c_0 - (p-1) = i + j + 1 + c_0 - p \geq 1,$$

y se tiene que

$$i + j + 1 + c_0 - p + k \leq 3l + p - l - 1 - c_0 + 1 + c_0 - p = 2l,$$

esto es,

$$\alpha_{i+j+c_0,k} = \alpha_{i+j+c_0-(p-1),k} = 0.$$

Por consiguiente, en ambos casos tenemos la igualdad $\alpha_{i,j}\alpha_{i+j+c_0,k} = 0$. □

Corolario 4.3.2. *Si $p \leq 2l + c_0 + 1$ y $p + 2l - c_0 - 1 \leq m - 2c - 1$, entonces $f(i, j, k) = 0$ siempre que $i + j + k \leq p + 2l + c_0 - 1$, esto es, se satisfacen las ecuaciones de Jacobi para $i + j + k \leq p + 2l - c_0 - 1$.*

Demostración. Es una consecuencia sencilla del Lema 4.3.1 aplicado a cada término de la expresión de la ecuación de Jacobi. □

En primer lugar, consideramos la fila $c_0 = p - 2$.

Lema 4.3.3. *Si $c_0 = p-2$, $l \geq 2$, entonces $2c \geq m-2l-p+c_0 = m-2l-2$. Más aún, existen álgebras de Lie de clase maximal con $2c = m - 2l - 2$.*

Demostración. Supongamos que $2l+2 \leq m-2c-1$. Si aplicamos la identidad de Jacobi para la terna $(1, l, l+1)$, tenemos que

$$\alpha_{1,l}\alpha_{l+1+c_0,l+1} + \alpha_{l,l+1}\alpha_{2l+1+c_0,1} + \alpha_{l+1,1}\alpha_{l+2+c_0,l} = 0,$$

y, aplicando la periodicidad módulo $p - 1$, tenemos:

$$\alpha_{1,l}\alpha_{l,l+1} + \alpha_{l,l+1}\alpha_{2l,1} + \alpha_{l+1,1}\alpha_{l+1,l} = 0.$$

Como $1 + l \leq 2l + 1$ and $l + 1 + 1 \leq 2l + 1$, obtenemos que

$$\alpha_{l,l+1}\alpha_{2l,1} = 0,$$

de donde $x_l = 0$, una contradicción.
Si $c_0 = p - 2$, $2l + c_0 + 1 = 2l + p - 1 \geq p$, luego estamos en las hipótesis del Corolario 4.3.2 y cualquier asignación de las x_i para $l \leq i \leq (p - 3)/2$ satisface las ecuaciones de Jacobi para los niveles no mayores que $2l + 1$. □

Lema 4.3.4. *Supongamos que $p-3 \geq c_0 \geq p-l$, entonces $2c \geq m-2l-p+c_0$. Existen álgebras de Lie de clase maximal con $2c = m - 2l - p + c_0$.*

Demostración. Supongamos lo contrario, esto es, $p + 2l - c_0 \leq m - 2c - 1$. Observamos que $2l + c_0 + 1 \geq 2l + p - l + 1 = p + l - 1 \geq p$, luego todas las relaciones de Jacobi para las ternas (i, j, k) con $i + j + k < p + 2l - c_0$ se anulan, en virtud del Corolario 4.3.2.
Consideremos la identidad de Jacobi para $(c_0-p+2l+3, p-c_0-2, p-c_0-1)$. Tenemos entonces:

$$\alpha_{c_0-p+2l+3,p-c_0-2}\alpha_{2l+c_0+1,p-c_0-1} + \alpha_{p-c_0-1,c_0-p+2l+3}\alpha_{2l+c_0+2,p-c_0-2} = 0,$$

y $2l + c_0 + 1 \geq 2l + p - l + 1 = p - l + 1 > p - 1$, luego $\alpha_{2l+c_0+1,p-c_0-1} = \alpha_{2l+c_0-p+2,p-c_0-1}$, y $\alpha_{2l+c_0+2,p-c_0-2} = \alpha_{2l+c_0-p+3,p-c_0-2}$. Por otro lado, tenemos que $p - c_0 - 1 \leq l - 1$, con lo que

$$\alpha_{2l+c_0-p+2,p-c_0-1} = -\alpha_{2l+c_0-p+3,p-c_0-2} \neq 0,$$

ya que $2l + c_0 - p + 2 + p - c_0 - 1 = 2l + 1$.
Por consiguiente,

$$\alpha_{p-c_0-2,c_0-p+2l+3} + \alpha_{p-c_0-1,c_0-p+2l+3} = 0.$$

Si lo expresamos en función de x_l, obtenemos:

$$0 = (-1 + c_0 - p + l + 2)x_l = (c_0 - p + l + 1)x_l,$$

y, al ser $c_0 \geq p - l$, $c_0 - p + l + 1 \neq 0$, esto es, $x_l = 0$, una contradicción.
Concluimos que $2c \geq m - 2l - p + c_0$ en este caso. Tenemos también que cualquier asignación de las x_i para $l \leq i \leq (p-3)/2$ satisface las identidades de Jacobi para los niveles no superiores a $p + 2l - c_0 - 1$. □

A continuación estudiamos la última columna $2l = p - 3$.

Lema 4.3.5. *Supongamos que* $2l = p - 3$ *y* $c_0 \geq 4$ *ó* $c_0 = 2$. *Entonces* $2c \geq m - p - 2l + c_0$.

Demostración. Supongamos lo contrario, esto es, $p + 2l - c_0 \leq m - 2c - 1$. Observamos que todas las relaciones de Jacobi para las ternas (i, j, k) con $i + j + k < p + 2l - c_0$ se anulan, por aplicación del Corolario 4.3.2. Por consiguiente, la primera relación de Jacobi no nula aparece para $i + j + k \geq p + 2l - c_0$.
Supongamos ahora que $c_0 \geq 2$ es par. Entonces podemos considerar Jacobi para $(l - c_0/2, l - c_0/2 + 1, p - 1)$, lo que nos da

$$\alpha_{l-c_0/2+1,p-1}\alpha_{l+c_0/2+p,l-c_0/2} + \alpha_{p-1,l-c_0/2}\alpha_{p+l-1+c_0/2,l-c_0/2+1} = 0,$$

esto es,

$$\alpha_{l-c_0/2+1,p-1} + \alpha_{l-c_0/2,p-1} = 0,$$

y, teniendo en cuenta la periodicidad módulo $p - 1$, es fácil ver que $\alpha_{i,p} = 0$ para $i < 2l$, luego $\alpha_{i,p-1} = \alpha_{1,p-1}$ para $i \leq 2l$ y concluimos que $\alpha_{1,p-1} = 0$. En consecuencia,

$$\alpha_{1,2l+2} = (-1)^l x_{l+1} + (-1)^{l-1}\binom{l+1}{2}x_l = 0$$

y

$$\alpha_{2,2l+2} = (-1)^{l-1} l x_{l+1} + (-1)^{l-2}\binom{l+1}{3}x_l = 0,$$

esto es, $x_l = 0$, una contradicción.

Si $c_0 \geq 5$ es impar, consideramos Jacobi para la terna $(l - c_0/2 + 1/2, l - c_0/2 + 3/2, p - 2)$. Esto da

$$\alpha_{l-c_0/2+3/2,p-2} + \alpha_{l-c_0+1/2,p-2} = 0. \tag{4.1}$$

Denotemos $A = \alpha_{l-c_0/2+3/2,p-2}$, $B = \alpha_{1,p-1}$. Por la periodicidad módulo $p-1$, tenemos que $\alpha_{i,p} = 0$ para $i < l$. Por lo tanto, $\alpha_{i,p-1} = x$ para $i < l$. Como $\alpha_{l-c_0/2+1/2} = A + x$, la ecuación (4.1) puede reescribirse como $2A + x = 0$, esto es, $x = -A/2$. Concluimos que $\alpha_{1,p-2} = A + (l - c_0/2 + 1/2)x = (l - c_0/2 - 3/2)x$ y $\alpha_{1,p-1} = x$. De este modo, tenemos las condiciones

$$(-1)^{l+1}\binom{l}{1}x_l = (l - c_0/2 - 3/2)x,$$

$$(-1)^l x_{l+1} + (-1)^{l-1}\binom{l+1}{2}x_l = x$$

y

$$(-1)^l l x_{l+1} + (-1)^{l-1}\binom{l+2}{3}x_l = 0,$$

esto es, $x_l = 0$, una contradicción. □

Lema 4.3.6. *Si $c_0 \geq 2p - 4l - 1$ y $p + 2l - c_0 \leq m - 2c - 1$, entonces $x_r = 0$ para $p - c_0 \leq r \leq l + [(p - c_0)/2] - 1$.*

Demostración. Consideremos, para $0 \leq k \leq l + [(c_0 - p)/2] - 1$, la ecuación de Jacobi para la terna

$$(2l + c_0 - p - 2k - 1, p - c_0 + k, p - c_0 + 1 - k).$$

En primer lugar, $2l + c_0 - p - 2k - 1 \geq 1$, en otro caso,

$$\begin{aligned} 1 &> 2l + c_0 - p - 2k - 1 \\ &\geq 2l + 2p - 4l - 1 - p - 2k - 1 \\ &= p - 2l - 2 - 2k \\ &\geq p - 2l - 2 \\ &\geq 1, \end{aligned}$$

una contradicción. Más aún,

$$(2l + c_0 - p - 2k - 1) + (p - c_0 + 1 - k) = 2l - k \leq 2l,$$

luego la ecuación de Jacobi puede escribirse como

$$x_{p-c_0+k}\alpha_{2p-c_0+1-2k,2l+c_0-p-2k-1=0}.$$

De cara a probar que $2p - c_0 + 1 - 2k \geq p$, basta probar que $2l + c_0 - p - 2k - 1 \leq 2l$, o, equivalentemente, $c_0 - p - 2k - 1 \leq 0$. Pero esto es cierto, ya que $c_0 - p - 2k - 1 \leq c_0 - p - 1 < 0$. Por consiguiente, el segundo factor no se anula, y concluimos que $x_{p-c_0-k} = 0$, de donde $x_r = 0$ para $p - c_0 \leq r \leq l + [(p - c_0)/2] - 1$. □

Lema 4.3.7. *Supongamos que* $c_0 \geq 2p-4l+1$. *Entonces* $2c \geq m-(p+2l-c_0)$.

Demostración. Supongamos que $p + 2l - c_0 \leq m - 2c - 1$.
Por aplicación del Corolario 4.3.2, si $i + j + k \leq p + 2l - c_0 - 1$, bastará considerar las ecuaciones de Jacobi para $i + j + k = p + 2l - c_0$.
Podemos aplicar el Lema 4.3.6 y podemos concluir que

$$p - c_0 \leq r \leq l + \left[\frac{p - c_0}{2}\right] - 1.$$

Denotemos

$$u = \left[l - \frac{c_0 - 1}{2}\right].$$

Por la periodicidad módulo $p - 1$ tenemos que

$$0 = \alpha_{i,u} = \alpha_{i,p-1+u} = \sum_{k=i}^{\left[\frac{p-2+u+i}{2}\right]} (-1)^{k-i}\binom{p - 2 + u - k}{k - i}x_k,$$

ya que $u \leq l$. Recordemos que si $p - c_0 \leq r \leq l + \frac{p-c_0}{2} - 1$, $x_r = 0$. Como $[(p - 2)/2] = (p - 3)/2$, tenemos que

$$\left[\frac{p - 2 - u + i}{2}\right] \leq \left[\frac{p - 2}{2} + u\right] = \frac{p - 3}{2} + \left[l - \frac{c_0 - 1}{2}\right] \leq l + \frac{p - c_0}{2} - 1,$$

luego todas estas ecuaciones pueden ser escritas en términos de x_l, $x_{l+1}, \ldots$, $x_{l+[(p-c_0)/2]-1}$ y, por consiguiente, en términos de of x_l, $x_{l+1}, \ldots$, x_{p-c_0-1}.

Más aún, $p - c_0 - l \leq l - \frac{c_0-1}{2}$, porque, de otro modo, si $p - c_0 - l > l - \frac{c_0-1}{2}$, tendríamos que $2p - 2c_0 > 4l - c_0 + 1$, esto es, $c_0 < 2p - 4l - 1 < c_0$, una contradicción.

Consideremos, para $1 \leq i < p - c_0 - l$,

$$\begin{aligned} 0 = \alpha_{i,p-1+u} &= \sum_{k=l}^{p-c_0-1} (-1)^{k-i} \binom{p-2+u-k}{k-i} x_k \\ &= \sum_{s=1}^{p-l-c_0} (-1)^{s+l-1-i} \binom{p-1+u-l-s}{s+l-1-i} x_{s+l-1} \\ &= \sum_{s=1}^{p-l-c_0} (-1)^{s+l-1-i} \binom{p-1+u-l-s}{p+u-2l-2s+i} x_{s+l-1}. \end{aligned}$$

Sabemos que el determinante de la matriz de coeficientes de este sistema es, salvo signo,

$$\det A = \frac{F(p-1+u-l, p+u-2l, p-l-c_0)}{F(p+u-2l, p+u-2l, p-l+c_0)},$$

donde

$$\begin{aligned} F(r,t,e) = &\prod_{1 \leq w \leq t-1} (r-w)^{\min(w, t-e, t-w)} \\ &\times \prod_{t-e+1 \leq w \leq t+e-3} (2r-w-1)^{\min\left(\left[\frac{e-t+w+1}{2}\right], \left[\frac{e+t-w-1}{2}\right]\right)}, \end{aligned}$$

ya que $p+u-2l \geq p-l-c_0$ (recordemos que $l \leq u + c_0 = l - [(-c_0-1)/2]$). Supongamos que el cociente es un múltiplo de p. Entonces $p \mid (p-1+u-l-w)$ para algún w con $1 \leq w \leq p+u-2l-1$, o $p \mid \big(2(p-1+u-l)-w-1\big)$ para algún w con $(p+u-2l)-(p-l-c_0)+1 \leq w \leq (p+u-2l)+(p-l-c_0)-3$, esto es, $[(c_0+3)/2] \leq w \leq 2p-2l+[(-c_0+1)/2]-c_0-3$. La primera posibilidad no se da, ya que

$$\begin{aligned} 0 &< p-1+u-l-(p+u-2l-1) \\ &= l \leq p-1+u-l-w \\ &\leq p-2+u-l \\ &= p + \left[\frac{-c_0+1}{2}\right] - 2 \\ &< p, \end{aligned}$$

y tampoco se da la segunda, ya que

$$\begin{aligned}
&2p-3+2\left[\frac{-c_0+1}{2}\right]-\left(2p-2l+\left[\frac{-c_0+1}{2}\right]-c_0-3\right)\\
&\quad=\left[\frac{-c_0+1}{2}\right]+2l+c_0\\
&\quad\leq 2p-3+2\left[\frac{-c_0+1}{2}\right]-\left(\left[\frac{-c_0+1}{2}\right]+c_0+1\right)\\
&\quad=2p-4+\left[\frac{-c_0+1}{2}\right]-c_0
\end{aligned}$$

y

$$\begin{aligned}
&\left[\frac{-c_0+1}{2}\right]+2l+c_0\\
&\qquad\geq\left[\frac{-c_0+1}{2}\right]+p-\frac{c_0}{2}+\frac{1}{2}+c_0\\
&\qquad=\left[\frac{c_0+1}{2}\right]+p-\frac{c_0}{2}+\frac{1}{2}\\
&\qquad\geq p+\frac{1}{2},
\end{aligned}$$

esto es, el factor considerado es mayor que p y menor que $2p$, luego no puede ser un múltiplo de p.

Por consiguiente, tenemos que $x_l = 0$, una contradicción. □

Lema 4.3.8. *Supongamos que* $2p-4l-1 \leq c_0 \leq 2p-4l+2$ *y* $l > \frac{p-1}{3}$. *Entonces* $2c \geq m-p-2l+c_0$.

Demostración. Supongamos que $p+2l-c_0 \leq m-2c-1$. Tenemos que $p \leq 2l+c_0+1$, ya que, de otro modo, se tendría que

$$c_0 \geq 2p-4l-1 > 4l+2c_0+2-4l-1 = 2c_0+1,$$

de donde $c_0 \leq -1$, una contradicción.

Por aplicación del Corolario 4.3.2, sabemos que para $i+j+k \leq p+2l-c_0-1$, las expresiones de Jacobi se anulan. Por consiguiente, basta considerar las ecuaciones de Jacobi para $i+j+k = p+2l-c_0$.

Por el Lema 4.3.6, tenemos que $x_r = 0$ para $p - c_0 \leq r \leq l + [(p - c_0)/2] - 1$. Definamos

$$\beta_{i,j} = \alpha_{i+l+[(p-c_0)/2]-1, j+l+[(p-c_0)/2]-1}$$

y

$$u_i = \beta_{i,i+1}.$$

Notemos que $u_i = x_{i+l+[(p-c_0)/2]-1}$. Obtenemos, por tanto, que si $p - c_0 - l - [(p-c_0)/2] + 1 \leq i \leq 0$, entonces $u_i = 0$, esto es, $u_i = 0$ para $-l + (p-1)/2 + 2 + [-c_0/2] \leq i \leq 0$.
Por otro lado, la periodicidad módulo $p-1$ nos da $\alpha_{i,p} = 0$ para $1 \leq i \leq 2l-1$, en particular, para $p - c_0 \leq i \leq 2l - 1$. Consecuentemente,

$$\beta_{i-l-[(p-c_0)/2]+1, p-l-[(p-c_0)/2]+1} = 0 \qquad \text{para } p - c_0 \leq i \leq 2l - 1,$$

y, así,

$$\beta_{i-l-[(p-c_0)/2]+p-c_0, p-l-[(p-c_0)/2]+1} = 0 \qquad \text{para } 1 \leq i \leq 2l - p - c_0,$$

de donde

$$\beta_{i-l+(p+1)/2-[(c_0+1)/2], 1-l+(p+1)/2-[(1-c_0)/2]} = 0 \qquad \text{para } 1 \leq i \leq 2l - p - c_0. \tag{4.2}$$

Tenemos también que

$$\beta_{i,j} = \sum_{k=i}^{[(i+j-1)/2]} (-1)^{k-i} \binom{j-1-k}{k-i} u_k.$$

Por consiguiente, teniendo en cuenta que $u_i = 0$ para $-l + (p-1)/2 + 2 + [-c_0/2] \leq i \leq 0$ y $[-c_0/2] = -[(c_0+1)/2]$, obtenemos que $u_k = 0$ para $1 - l + (p+1)/2 - [(c_0+1)/2] \leq k \leq 0$.
El número de variables que aparecen en (4.2) es

$$\left[\frac{\big(2l - p + c_0 - l - (p+1)/2 - [(c_0+1)/2]\big) + \big(1 - l + (p+1)/2 - [(1-c_0)/2]\big)}{2} \right] =$$
$$= \left[\frac{c_0 - [(c_0+1)/2] - [(1-c_0)/2]}{2} \right],$$

que toma el valor$[c_0/2] = (c_0 - 1)/2$ si c_0 es impar, y $[(c_0+1)/2] = c_0/2 = [c_0/2]$ si c_0 es par.

Si c_0 es impar, como $c_0 \geq 2p-4l-1$, se sigue que $(c_0-1)/2 \leq 2l-p+c_0$, y si c_0 es par, entonces $c_0 \geq 2p-4l$, luego $c_0/2 \leq 2l-p+c_0$. En ambos casos, el número de incógnitas que aparece no es superior al número de ecuaciones. Podemos escribir las ecuaciones (4.2) como sigue:

$$\beta_{i-l+(p+1)/2-[(c_0+1)/2],1-l+(p+1)/2-[(1-c_0)/2]}$$
$$\sum_{k=1}^{[c_0/2]}(-1)^{k-i+l-(p+1)/2+[(c_0+1)/2]}\binom{(p+1)/2-l-[(1-c_0)/2]-k}{k-i+l-(p+1)/2+[(c_0+1)/2]}u_k$$

para $1 \leq i \leq 2l-p+c_0$. En particular, podemos escribir esto para $1 \leq i \leq [c_0/2]$. Tenemos también la siguiente igualdad:

$$\binom{(p+1)/2-l-[(1-c_0)/2]-k}{k-i+l-(p+1)/2+[(c_0+1)/2]} = \binom{(p+1)/2-l-[(1-c_0)/2]-k}{p-2l-[(1-c_0)/2]-[(c_0-1)/2]-2k+i}.$$

Sea

$$r = \frac{p+1}{2} - l - \left[\frac{1-c_0}{2}\right],$$
$$t = p-2l-\left[\frac{1-c_0}{2}\right]-\left[\frac{c_0-1}{2}\right],$$
$$e = \left[\frac{c_0}{2}\right].$$

Para c_0 par, $t = p-2l+1$, y, para c_0 impar, $t = p-2l$. Si c_0 es par $t \geq e$, ya que $p-2l+1 \geq c_0/2$ (recordemos que $c_0 \leq 2p-4l+2$), y si c_0 es impar, $t \geq e$, esto es, $p-2l \geq (c_0-1)/2$, ya que $c_0 \leq 2p-4l+1$. Por tanto, $t \geq e$. El determinante de este sistema no se anula. En efecto, para $1 \leq w \leq t-1$, $r-w = (p+1)/2-l-[(1-c_0)/2]-w$, de donde

$$\frac{-p+1}{2}+l+\left[\frac{c_0-1}{2}\right]+1 \leq r-w \leq \frac{p-1}{2}-l-\left[\frac{1-c_0}{2}\right]+\left[\frac{c_0}{2}\right]$$

y, así,

$$\begin{aligned}0 &< \frac{p+1}{2}-l\\ &= \frac{-p+1}{2}+l+p-2l-1+1 \leq r-w \leq \frac{p-1}{2}-l-\left[\frac{-1-2p+4l}{2}\right]\\ &= \frac{p-1}{2}-l+1+p-2l\\ &= p+\frac{p-1}{2}-3l+1\end{aligned}$$

y $(p+1)/2 - 3l < 0$ ya que $l > (p+1)/6$, y para $t - e + 1 \leq w \leq t + e - 3$, donde $t - e + 1 = p - 2l - [(1-c_0)/2] - [(c_0-1)/2] - [c_0/2] + 1$ y $t + e - 3 = p - 2l - [(1-c_0)/2] - [(c_0-1)/2] - 3 + [c_0/2]$, el número $2r - 1 - w = p + 1 - 2l - 2[(1-c_0)/2] - w$ está comprendido entre

$$\begin{aligned} 4&-[(1-c_0)/2] + [(c_0-1)/2] - [c_0/2] \\ &\geq 4 - [(2-2p+4l)/2] + [(2p-4l-2)/2] - [(2p-4l-2)/2] \\ &= 4 - 1 + p - 2l \\ &= p - 2l + 3 \end{aligned}$$

y

$$\begin{aligned} -[(1-c_0)/2] &+ [(c_0-1)/2] + [c_0/2] \\ &\leq -[(-2p+4l-2)/2] + [(2p-4l)/2] + [(2p-4l+2)/2] \\ &= p - 2l + 1 + p - 2l + p - 2l + 1 \\ &= 3p - 6l + 2 \\ &< 3p - 2p - 2 + 2 \\ &= 2p - 1, \end{aligned}$$

luego el determinante no puede anularse.

Por consiguiente, tenemos que $u_1 = u_2 = \cdots = u_{[c_0/2]}$. En particular, ningún elemento de la columna $p - c_0$ es diferente de cero, luego todo el triángulo debe ser cero, una contradicción. □

A continuación, estudiamos la diagonal $l + c_0 = (p+1)/2$.

Lema 4.3.9. *Supongamos que* $l + c_0 = (p+1)/2$, $4l + c_0 - 1 \leq m - 2c - 1$ *y* $l \geq 2$. *Entonces* $\alpha_{i,2l+c_0} = 0$ *para* $1 \leq i \leq 2l - 3$.

Demostración. Consideremos la ecuación de Jacobi para $(i, 2l-2-i, 2l+c_0-1)$, donde $1 \leq i \leq l-1$. Tenemos

$$\alpha_{2l-2-i,2l+c_0-1}\alpha_{4l+2c_0-1-i,i} + \alpha_{i,2l+c_0+1}\alpha_{2l-2-i,2l+2c_0+i+1} = 0,$$

Teniendo en cuenta que $l + c_0 = (p+1)/2$, tenemos que $2l + 2c_0 = p + 1$, luego

$$\alpha_{2l-2-i,2l+c_0-i}\alpha_{2l+1-i,i} + \alpha_{i,2l+c_0+1}\alpha_{2l-2-i,i+3}.$$

Por consiguiente,

$$\alpha_{2l-2-i,2l+c_0-i} = \alpha_{i,2l+c_0+1}.$$

Pero sabemos, por el Lema 4.2.1, que $\alpha_{i,2l+c_0+1} = 0$ for $1 \leq i \leq l-1$. Luego $\alpha_{2l-2-i,2l+c_0+1} = 0$ para $1 \leq i \leq l-1$. Por tanto,

$$\alpha_{i,2l+c_0+1} = 0 \qquad \text{para } 1 \leq i \leq 2l-3. \tag{4.3}$$

Consideremos la terna $(1, 2l-2, 2l+c_0)$. Obtenemos

$$\begin{aligned} 0 &= \alpha_{2l-2,2l+c_0}\alpha_{4l+2c_0-2,1} + \alpha_{2l+c_0,1}\alpha_{2l+2c_0+1,2l-2} \\ &= \alpha_{2l-2,2l+c_0}\alpha_{2l,1} + \alpha_{2l+c_0,1}\alpha_{3,2l-1}, \end{aligned}$$

luego

$$\alpha_{2l-2,2l+c_0}(-\alpha_{1,2l}) + \alpha_{1,2l+c_0}(-\alpha_{1,2l}) = 0,$$

por tanto

$$\alpha_{2l-2,2l+c_0} + \alpha_{1,2l+c_0} = 0.$$

Pero por (4.3), tenemos que $\alpha_{i,2l+c_0} = \alpha_{1,2l+c_0}$ para $1 \leq i \leq 2l-2$, luego

$$2\alpha_{1,2l+c_0} = 0,$$

por tanto $\alpha_{1,2l+c_0} = 0$ y tenemos el resultado deseado. □

Teorema 4.3.10. *Supongamos que $l \geq (p+7)/6$ y $l + c_0 = (p+1)/2$. Entonces $2c \geq m - p - 2l + c_0$.*

Demostración. Por reducción al absurdo, supongamos que se da lo contrario, es decir, $p + 2l - c_0 = 4l + c_0 - 1 \leq m - 2c - 1$. Por el Lema 4.3.9, obtenemos que $\alpha_{i,2l+c_0+1} = 0$ para $1 \leq i \leq 2l-3$. Como $l \geq (p+7)/6$, entonces $3l \geq (p+1)/2 + 3$, luego $2l \geq (p+1)/2 - l + 3 = c_0 + 3$. En consecuencia, tenemos que $\alpha_{i,2l+c_0} = 0$ para $1 \leq i \leq c_0$.

Podemos escribir, usando números combinatorios generalizados,

$$\begin{aligned} \alpha_{i,2l+c_0} &= \sum_{k=l}^{l+c_0-1} (-1)^{k-i} \binom{2l+c_0-k-1}{k-i} x_k \\ &= \sum_{j=1}^{c_0} (-1)^{j+l-1-i}(-1)^{j+l-1-i} \binom{2l+c_0-(j+l-1)-1}{(j+l-1)-i} x_{j+l-1}, \end{aligned}$$

un sistema de c_0 ecuaciones con c_0 incógnitas cuya matriz de coeficientes es, salvo signo, $(a_{i,j})$, con

$$a_{i,j} = \binom{2l + c_0 - j - l + 1 - 1}{j + l - i - 1} = \binom{l + c_0 - j}{j + l - i - 1} = \binom{l + c_0 - j}{c_0 + 1 - 2j + i}.$$

Llamemos $r = l + c_0$, $t = c_0 + 1$, $e = c_0$. Sabemos, por [28, Lemma 1], que esta matriz es singular si, y sólo si, $p \mid F(r, t, e)$, donde

$$F(r,t,e) = \prod_{1 \leq w \leq t-1} (r-w)^{\min(w, t-e, t-w)} \cdot \prod_{t-e+1 \leq t+e-3} (2r-w-1)^{\min([\frac{e-t+w-1}{2}], [\frac{e+t-w-1}{2}])}.$$

Pero p no puede dividir ningún factor of $F(r, t, e)$, ya que si $1 \leq w \leq t - 1 = c_0$, tenemos que $0 < l \leq r - w \leq l + c_0 - 1 < p$, y si $2 = t - e + 1 \leq w \leq t + e - 3 = 2c_0 - 2$, tenemos que $2r - w - 1 = 2l + 2c_0 - 1 - w$, luego

$$0 < 2l + 1 = 2l + 2 - 1 \leq 2r - w - 1 \leq 2l + 2c_0 - 3 = p - 2 < p.$$

Por tanto, el determinante de esta matriz no se anula, en particular, el sistema tiene una solución única, $x_i = 0$ para $l \leq i \leq l + c_0 - 1$, en contradicción con la condición $x_l \neq 0$. □

A continuación intentamos estudiar el caso $c_0 = 2p - 4l - 2$.

Lema 4.3.11. *Si $c_0 \leq 2l - 2$ y $2l - c_0 + p \leq m - 2c - 1$, entonces*

$$\alpha_{2l-c_0+i, p-1-i} = (-1)^{c_0 - i} \alpha_{p-1-i, 1}$$

para $0 \leq i \leq c_0 - 1$.

Demostración. Para $0 \leq i \leq c_0 - 1$, podemos considerar las ternas $(1, 2l - c_0 + i, p - 1 - i)$, entonces la condición de Jacobi nos da

$$\begin{aligned} 0 &= \alpha_{2l-c_0+i, p-1-i} \alpha_{2l,1} + \alpha_{p-1-i,1} \alpha_{p-1-i+c_0+1, 2l-c_0+i}, \\ &= \alpha_{2l-c_0+i, p-1-i}(-y_0) + \alpha_{p-1-i,1}(-1)^{c_0-i} y_0, \end{aligned}$$

luego

$$\alpha_{2l-c_0+i, p-1-i} = (-1)^{c_0-i} \alpha_{p-1-i,1}. \tag{4.4}$$

□

Lema 4.3.12. *Si* $c_0 \leq 2l-2$, $2l-c_0+p \leq m-2c-1$ *y* $c_0 > p-2l+1$, *entonces:*

1. $$x_{p-c_0} = \cdots = x_{l-1+[(p-c_0)/2]} = 0. \tag{4.5}$$

2. $x_{p-c_0-1} = (-1)^{l+c_0} x_l$ *y* $x_{p-c_0-2} = (-1)^{l+c_0}\big(l-(p-c_0)+1\big)x_l$.

Demostración. 1. Consideremos en (4.4) $p-2l \leq i \leq c_0-1$. Entonces $p-1-i+1 \leq p-1-(p-2l)+1 = 2l$, por tanto

$$\alpha_{2l-c_0+i,p-1-i} = 0 \qquad \text{para } p-2l \leq i \leq c_0-1. \tag{4.6}$$

Los elementos $\alpha_{p-c_0,2l-1}$, $\alpha_{p-c_0+1,2l-2}, \ldots, \alpha_{2l-1,p-c_0}$ de la fila $p+2l-c_0-1$ son todos cero, y uno de ellos es x_u. Si $p-c_0$ es par, este u es tal que $2u+1 = p-c_0+2l-1$, o sea, $u = l-1+(p-c_0)/2$, y si $p-c_0$ es impar, entonces u es tal que $2u+2 = p-c_0+2l-1$, esto es, $u = l-1+(p-c_0-1)/2$. En cualquier caso, $u = l-1+[(p-c_0)/2]$. Teniendo en cuenta las propiedades (P1) y (P3), (4.6) es equivalente a

$$x_{p-c_0} = \cdots = x_{l-1+[(p-c_0)/2]} = 0.$$

2. Si tomamos en (4.4) $i = p-2l-1$, tenemos que

$$\alpha_{p-c_0-1,2l} = (-1)^{c_0-1}\alpha_{1,2l} = x_l.$$

Más aún, como $\alpha_{p-c_0-1,j} = \alpha_{p-c_0-1,2l}$ for $p-c_0 \leq j \leq 2l$, teniendo en cuenta (4.5), se llega a que $x_{p-c_0-1} = (-1)^{l+c_0}x_l$. Si consideramos $i = p-2l-2$, obtenemos de (4.4) la igualdad $\alpha_{p-c_0-2,2l+1} = (-1)^{c_0}\alpha_{1,2l+1} = (-1)^{c_0}ly_0 = (-1)^{l+c_0-1}lx_l$. Tenemos que $\alpha_{p-c_0-2,2l+1-j} = \alpha_{p-c_0-2,2l+1} + (-1)^{l+c_0}jx_l$ para $2l+1-j \geq p-c_0$, luego para $j = 2l+1-(p-c_0)$, obtenemos

$$\begin{aligned}
\alpha_{p-c_0-2,p-c_0} &= x_{p-c_0-2} \\
&= \alpha_{p-c_0-2,2l+1} + (-1)^{l+c_0}(2l+1+c_0-p)x_l \\
&= (-1)^{l+c_0-1}lx_l + (-1)^{l+c_0}(2l+1+c_0-p)x_l \\
&= (-1)^{l+c_0}(l+1+c_0-p)x_l.
\end{aligned}$$

□

Lema 4.3.13. *Supongamos que c_0 es par, $c_0 < 2l$ y $p+2l-c_0 \leq m-2c-1$, entonces $\alpha_{1,p-1} = 0$.*

Demostración. Consideremos la terna $(1, p-1, 2l-c_0)$, que nos lleva a

$$\begin{aligned} 0 &= \alpha_{1,p-1}\alpha_{1+p-1+c_0,2l-c_0} + \alpha_{p-1,2l-c_0}\alpha_{2l+p-1,1} + \alpha_{2l-c_0,1}\alpha_{2l+1,p-1} \\ &= \alpha_{1,p-1}\alpha_{1+c_0,2l-c_0} + \alpha_{p-1,2l-c_0}\alpha_{2l,1} \\ &= \alpha_{1,2l}\big(\alpha_{1,p-1}(-1)^{c_0} + \alpha_{2l-c_0,p-1}\big), \end{aligned}$$

y, teniendo en cuenta que $\alpha_{1,p-1} = \alpha_{2l-c_0,p-1}$ y que c_0 es par, llegamos a $\alpha_{1,p-1} = 0$, como deseábamos. □

Lema 4.3.14. *Supongamos que $p \leq 3l-1$ y $c_0 = 2p-4l-2$, entonces $2c \geq m-p-2l+c_0$.*

Demostración. Notemos que si $c_0 = 2p-4l-2$ y $p \leq 3l-1$, $2p \leq 6l-2 \leq 6l$, por tanto $c_0 = 2p-4l-2 \leq 2l-2$, luego por el Lema 4.3.13 tenemos que $\alpha_{i,p-1} = 0$ para $1 \leq i < 2l$. Por (4.5), también sabemos que $x_k = 0$ para $p - c_0 \leq k \leq l - 1 + [(p-c_0)/2]$.
Para $1 \leq i \leq p-l-c_0$, consideremos $\alpha_{i+3l-1-p,p-1} = 0$ (recordemos que $i+3l-1-p \leq p-l-c_0+3l-1-p = 2l-c_0-1 < 2l$).

$$\begin{aligned} 0 &= \alpha_{i+3l-1-p,p-1} \\ &= \sum_{k=1}^{[(i+3l-3)/2]} (-1)^{k-i-3l+1+p}\binom{p-2-k}{k-i-3l+1+p}x_k. \end{aligned}$$

Notemos que $k \leq [(p-l-c_0+3l-3)/2] \leq l-1+[(p-c_0)/2]$, ya que $[(p-c_0)/2] = (p-c_0-1)/2$. Por consiguiente,

$$\begin{aligned} 0 &= \sum_{k=l}^{p-c_0-1} (-1)^{k-i-3l+1+p}\binom{p-2-k}{k-i-3l+1+p}x_k \\ &= \sum_{j=1}^{p-c_0-l} (-1)^{j-i-2l+p}\binom{p-l-1-j}{p-2l+j-i}x_{j+l-1} \\ &= \sum_{j=1}^{p-c_0-l} (-1)^{j-i-2l+p}\binom{p-l-1-j}{l-1-2j+i}x_{j+l-1}. \end{aligned}$$

Llamemos $r = p - l - 1$, $t = l - 1$, $e = p - l - c_0$. Notemos que $t \geq e$, pues $c_0 = 2p - 4l - 2$ y $p \geq 2l + 3$. El determinante de la matriz de este sistema es, salvo signo,

$$\frac{F(r,t,e)}{F(t,t,e)},$$

donde

$$F(r,t,e) = \prod_{1 \leq w \leq t-1} (r-w)^{\min(w,t-e,t-w)} \cdot \prod_{t-e+1 \leq w \leq t+e-3} (2r-w-1)^{\min([\frac{e-t+w+1}{2}],[\frac{e+t-w-1}{2}])}.$$

Si este determinante se anula, tendríamos que p divide $r-w$ para algún w con $1 \leq w \leq t-1$, ó p divide $2r-w-1$ para algún w con $t-e+1 \leq w \leq t+e-3$. Pero si $1 \leq w \leq t-1 = l-2$, entonces $p-2l+1 \leq r-w \leq p-l-2$, luego p no divide $r-w$, y si $2l+c_0-p = t-e+1 \leq w \leq t+e-3 = p-c_0-4$ y $c_0 = 2p-4l-2$, $p-2l-2 \leq w \leq 4l-p-2$, $3p-6l-1 \leq 2p-2l-3-w = 2r-w-1 \leq p-1$, y $3p-6l-1 > 0$, ya que $2l \leq p-3$, por tanto, p no divide $2r-w-1$. En consecuencia, este determinante no es divisible por p, luego llegamos a la contradicción $x_l = 0$. □

No hemos sido capaces de probar la cota $2c \geq m-p-2l+c_0$ para el caso $c_0 = 2p-4l-4$, $3l > p$, $l \leq (p-5)/2$. En la Sección 4.6 mostramos que la cota $2c \geq m-p-2l+c_0-1$ se da para estos valores. Evidentemente, la desigualdad es estricta para cada exponente m tal que $m \not\equiv -2l \pmod{p-1}$.

El siguiente teorema nos sugiere que la cota conjeturada será correcta en general.

Consideremos la matriz $e \times e$ $A = (a_{i,j})$, con $e = p-2l-2$, por la regla

$$a_{i,j} = \begin{cases} \binom{\frac{3}{2}(p-2l-1)-1-i}{p-2l-2i+j} & \text{para } 1 \leq j \leq e-1, \\ \binom{\frac{3}{2}(p-2l-1)-3-i}{i-3} & \text{para } j = e. \end{cases}$$

Teorema 4.3.15. *Supongamos que $3l > p$ y $c_0 = 2p-4l-4$, tenemos la cota $2c \geq m-p-2l+c_0$ siempre que se dé una de las siguientes condiciones:*

1. $\alpha_{1,p-2} = 0$.

2. $p \nmid \det A$.

3. $p \nmid \det B + (-1)^{l-1} \det C$, *siendo* $B = (b_{i,j})$ *la matriz dada por*

$$b_{i,j} = \binom{p-1-j-l}{l-2j+i}$$

para $1 \leq i, j \leq p - c_0 - l - 1$, *y* $C = (c_{i,j})$ *dada por*

$$c_{i,j} = \binom{p-2-j-l}{l-2-2j+i}.$$

Demostración. Si $l = (p-3)/2$, entonces $c_0 = 2$ y $c_0 + l = (p+1)/2$, luego tenemos la cota $2c \geq m - p - 2l + c_0$. Supongamos $l \leq (p-5)/2$.

1. Supongamos que $\alpha_{1,p-2} = 0$.

   Notemos que si $c_0 = 2p - 4l - 4$ y $p \leq 3l - 1$, $2p \leq 6l - 2$, luego $c_0 = 2p - 4l - 2 \leq 2l - 2$, luego por Lema 4.3.13 tenemos que $\alpha_{i,p-1} = 0$ para $1 \leq i \leq 2l$, y por hipótesis tenemos que $\alpha_{i,p-1} = 0$ para $1 \leq i \leq 2l + 1$. Por (4.5) también sabemos que $x_k = 0$ para $p - c_0 \leq k \leq l - 1 + [(p - c_0)/2]$.

   Para $1 \leq i \leq p - l - c_0$, podemos considerar $\alpha_{i+3l-p,p-2} = 0$ (recordemos que $i + 3l - p \leq p - l - c_0 + 3l - p = 2l - c_0 \leq 2l$).

$$\begin{aligned} 0 &= \alpha_{i+3l-p,p-2} \\ &= \sum_{k=1}^{[(i+3l-3)/2]} (-1)^{k-i-3l+p} \binom{p-3-k}{k-i-3l+p} x_k. \end{aligned}$$

   Notemos que $k \leq [(p - l - c_0 + 3l - 3)/2] \leq l - 1 + [(p - c_0)/2]$, ya que $[(p - c_0)/2] = (p - c_0 - 1)/2$. Por consiguiente,

$$\begin{aligned} 0 &= \sum_{k=l}^{p-c_0-1} (-1)^{k-i-3l+p} \binom{p-3-k}{k-i-3l+p} x_k \\ &= \sum_{j=1}^{p-l-c_0} (-1)^{j-2l+p-1-i} \binom{p-2-l-j}{p-2l-1+j-i} x_{j+l-1} \\ &= \sum_{j=1}^{p-l-c_0} (-1)^{j-2l+p-1-i} \binom{p-2-l-j}{l-1-2j+i} x_{j+l-1}. \end{aligned}$$

Denotemos $r = p-l-2$, $t = l-1$ y $e = p-l-c_0$. Observemos que $t \geq e$, ya que $c_0 = 2p - 4l - 4$ y $2l + 5 \leq p$. Entonces el determinante de la matriz de coeficientes de este sistema es, salvo signo, $F(r,t,e)/F(t,t,e)$. Si este determinante se anulara, tendríamos que p dividiría $r - w$ para algún w con $1 \leq w \leq t-1$, ó p dividiría $2r - w - 1$ para algún w con $t - e + 1 \leq w \leq t + e - 3$. Pero si $1 \leq w \leq t - 1 = l - 2$, entonces $p - 2l + 1 \leq r - w \leq p - l - 2$, luego p no dividiría $r - w$, y si $2l + c_0 - p = t - e + 1 \leq w \leq t + e - 3 = p - c_0 - 4$ y $c_0 = 2p - 4l - 4$, $3p - 6l - 3 \leq 2p - 2l - 5 - w \leq 2r - 1 - w \leq p - 1$ y $3p - 6l - 1 > 0$, ya que $2l \leq p - 3$. En consecuencia, este determinante no es divisible por p, llegaríamos a la contradicción $x_l = 0$.

2. En primer lugar, observemos que tenemos la igualdad $(2l + c_0 - 1)t_1 + 2t_2 = 0$, con $t_i = \alpha_{2l+1,p-1-k}$ para $p - 2l - c_0 \leq m - 2c - 1$. En efecto, es sabido que $\alpha_{j,p-3} = (2l+1-j)t_1 + t_2$ para $j \leq 2l+1$. Por tanto, Jacobi aplicado a $(i, p-3, 2l-c_0-i+3)$ da la condición $(2l+c_0-1)t_1+2t_2 = 0$. Por consiguiente, si $l + 1 + \frac{p-c_0+1}{2} + i \leq 2l + 2$, entonces

$$\alpha_{l+1+\frac{p-c_0+1}{2}-i,p-3} + \alpha_{l+1+\frac{p-c_0+1}{2}+i,p-3} = 0,$$

y, en particular, $\alpha_{l+1+\frac{p-c_0+1}{2},p-3} = 0$, ya que tenemos que

$$\alpha_{l+1+\frac{p-c_0+1}{2}-i} = \left(l - \frac{p - c_0 + 1}{2} + i\right) t_1 + t_2$$

y

$$\alpha_{l+1+\frac{p-c_0+1}{2}+i} = \left(l - \frac{p - c_0 + 1}{2} - i\right) t_1 + t_2,$$

luego su suma es

$$(2l - p + c_0 - 1)t_1 + t_2 = (2l + c_0 - 1)t_1 + 2t_2 = 0,$$

y la última afirmación se sigue tomando $i = 0$.

Consideremos, para $1 \leq i \leq p - 2l - 3$,

$$\begin{aligned}
0 &= \alpha_{p-c_0-1+i,p-1} \\
&= \alpha_{-p+4l+3+i,p-1} \\
&= \sum_{k=-p+4l+3+i}^{[(4l+1+i)/2]} (-1)^{k+p-4l-3-i} \binom{p-2-k}{k+p-4l-3-i} x_k \\
&= \sum_{s=-p+l+3+i+\frac{p-1}{2}}^{\left[\frac{4l+1+i}{2}\right]-3l+\frac{p-1}{2}} (-1)^{s-l+\frac{p+1}{2}-3-i} \binom{p-2-s-3l+\frac{p-1}{2}}{s-l+\frac{p+1}{2}+p-i} x_{s+3l+\frac{1-p}{2}}
\end{aligned}$$

y, teniendo en cuenta que $x_k = 0$ para $4l - p + 4 = p - c_0 \leq k \leq l - 1 + (p - c_0 - 1)/2 = (6l - p + 1)/2$ y que $\left[\frac{4l+1+i}{2}\right] - 3l + \frac{p-1}{2} \leq p - 2l - 2$ para $i \leq p - 2l - 3$, tenemos

$$\begin{aligned}
0 &= \sum_{s=1}^{p-2l-2} (-1)^{s-l+\frac{-p+1}{2}+p-3-i} \binom{p-2-s-3l+\frac{p-1}{2}}{s-l+\frac{-p+1}{2}+p-3-i} x_{s+3l+\frac{1-p}{2}} \\
&= \sum_{s=1}^{p-2l-2} (-1)^{s-l+\frac{-p+1}{2}+p-3-i} \binom{\frac{3p-3-6l}{2}-1-s}{-l+\frac{p-1}{2}-3+s-i} x_{s+3l+\frac{1-p}{2}} \\
&= \sum_{s=1}^{p-2l-2} (-1)^{s+3l+\frac{-p+1}{2}+p-4l-3-i} \binom{\frac{3}{2}(p-2l-1)-1-s}{p-2l-2s+i} x_{s+3l+\frac{1-p}{2}}.
\end{aligned}$$

Por otro lado, tenemos que $\alpha_{l+1+(p-c_0+1)/2,p-3} = \alpha_{(6l-p+7)/2,p-3} = 0$, y

$$\begin{aligned}
0 &= \alpha_{(6l-p+7)/2,p-3} \\
&= \sum_{k=(6l-p+7)/2}^{p-3} (-1)^{k-(6l-p+7)/2} \binom{p-4-k}{k-(6l-p+7)/2} x_k \\
&= \sum_{s=1}^{p-2l-2} (-1)^{s-3} \binom{p-4-s-3l+(p-1)/2}{s-3} x_{s+3l+(-p+1)/2} \\
&= \sum_{s=1}^{p-2l-2} (-1)^{s-3} \binom{\frac{3}{2}(p-2l-1)-3-s}{s-3} x_{s+3l+(-p+1)/2}.
\end{aligned}$$

Si p no dividiera $|A|$, obtendríamos, en particular, que $\alpha_{p-c_0,p-2} = 0$ y, por tanto, $\alpha_{1,p-2} = 0$. Por aplicación de la parte 1 obtendríamos la cota deseada.

3. Supongamos que $p \leq 3l-1$, $l \leq (p-5)/2$ y $c_0 = 2p-4l-4$. Para $1 \leq i \leq p-c_0-l-1$, podemos considerar $\alpha_{i+3l-p,p-1} = 0$ (observemos que $i+3l-p \leq p-c_0-l-1+3l-p = 2l-c_0 \leq 2l$). En primer lugar, tenemos que

$$\left[\frac{(i+3l-p)+(p-1)-1}{2}\right] \leq \left[\frac{(p-c_0-l-1+3l-p)+(p-1)-1}{2}\right] \leq l-1+\left[\frac{p-c_0}{2}\right],$$

teniendo en cuenta que $[(p-c_0)/2] = (p-c_0-1)/2$, luego podemos expresar

$$\begin{aligned} 0 &= \alpha_{i+3l-p,p-1} \\ &= \sum_{k=l}^{p-c_0-1} (-1)^{k-i-3l+p} \binom{p-2-k}{k-i-3l+p} x_k \\ &= \sum_{j=1}^{p-c_0-l} (-1)^{j-1-i-2l+p} \binom{p-1-j-l}{j-2l-i-1+p} x_{j+l-1} \\ &= \sum_{j=1}^{p-c_0-l} (-1)^{j-1-i-2l+p} \binom{p-1-j-l}{l-2j+i} x_{j+l-1}, \end{aligned}$$

Por otro lado, el Lema 4.3.12 nos da la condición

$$x_{p-c_0-1} = (-1)^{c_0+l} x_l,$$

luego podemos escribir

$$\begin{aligned}
0 = &\left((-1)^{p-2l-i}\binom{p-2-l}{l-2+i} \right. \\
&\qquad \left. + (-1)^{p-c_0-l-1-i-2l+p}(-1)^{c_0+l}\binom{p-1-(p-c_0-l)-l}{l-2(p-c_0-l)+i} \right) x_l \\
&+ \sum_{j=2}^{p-c_0-l-1} (-1)^{j-1-i-2l+p}\binom{p-1-j-l}{l-2j+i} x_{j+l-1}. \\
= (-1)^{p-2l-i} &\left(\binom{p-2-l}{l-2+i} + \binom{p-1-(p-c_0-l)-l}{l-2(p-c_0-l)+i} \right) x_l \\
&+ \sum_{j=2}^{p-c_0-l-1} (-1)^{j-1-i-2l+p}\binom{p-1-j-l}{l-2j+i} x_{j+l-1}.
\end{aligned}$$

Denotemos

$$d_{i,j} = \binom{p-1-j-l}{l-2j+i}.$$

Debemos mostrar que el determinante de este sistema, que es, salvo signo,

$$|D| = \begin{vmatrix} d_{1,1}+d_{1,p-c_0-l} & d_{1,2} & \dots & d_{1,p-c_0-l-1} \\ d_{2,1}+d_{2,p-c_0-l} & d_{2,2} & \dots & d_{2,p-c_0-l-1} \\ \vdots & \vdots & \ddots & \vdots \\ d_{p-c_0-l-1,1}+d_{p-c_0-l-1,p-c_0-l} & d_{p-c_0-l-1,2} & \dots & d_{p-c_0-l-1,p-c_0-l-1} \end{vmatrix},$$

no es divisible por p. Consideremos la matriz $B = (b_{i,j})$ dada por

$$b_{i,j} = d_{i,j}$$

para $1 \leq i, j \leq p - c_0 - l - 1$, y la matriz $C = (c_{i,j})$ dada por

$$c_{i,j} = d_{i,j+1}.$$

Tenemos que

$$|D| = |B| + (-1)^{p-c_0-l-1-1}|C| = |B| + (-1)^{l+1}|C|.$$

Como p no divide el segundo miembro, p no divide $|D|$.

□

Nota 4.3.16. Supongamos que $2l + 5 \leq p \leq 3l - 1$.

1. Si expresamos $p = 2l+a$, podemos calcular $\det A$ para valores pequeños valores de a. Calculando estos valores con la ayuda de GAP (véase [21]) obtenemos que los primos que aparecen en la factorización de $\det A$ no son mayores que $3a - 10$ para $a = p - 2l \leq 121$, en particular, estos primos no exceden $p - 10$ para $p < 367$ y, para estos valores de p, se tiene la segunda hipótesis.

2. Es posible calcular los determinantes de B y C con la fórmula (2.29). Por desgracia, el cálculo de este determinante es muy tedioso desde el punto de vista computacional, ya que los primos que aparecen en él son menores o iguales que l o muy grandes en relación a $3l$. Hemos verificado esto para $p = 3l-b$ con $1 \leq b \leq 8$ y hemos observado que los números primos que aparecen en esta suma pertenecen a $(2, l)\cup(3l, \infty)$. Por tanto, también se da la tercera hipótesis.

4.4. Cotas del tipo $2c \geq m - p - 1$

Lema 4.4.1. *Supongamos que $l + c_0 = p - 1$, $2 \leq l \leq p/3$. Entonces $2c \geq m - p - 1$. Además, la asignación $x_{l+1} = \cdots = x_{(p-3)/2} = 0$ satisface $\mathcal{S}(p)$.*

Demostración. Supongamos que $p + 1 \leq m - 2c - 1$, $l \geq 3$. Para $1 \leq k \leq (l - 1)/2$, $1 \leq k \leq (p - 1)/2 - l$, consideramos Jacobi para la terna $(l - 2k, l + k, l + k + 1)$. Teniendo en cuenta que $\alpha_{l-2k,l+k} = \alpha_{l+k+1,l-2k} = 0$, $\alpha_{l+2k+1,l-2k} \neq 0$, obtenemos que $\alpha_{l+k,l+k+1} = x_{l+k} = 0$.
Por consiguiente, tenemos que $x_{l+1} = x_{l+2} = \cdots = x_{l+[(l+1)/2]} = 0$.
Si $l > 3$, $(3l-1)/2 < (p-1)/2$, entonces $l+[(l-1)/2] < (p-1)/2$ y podemos considerar Jacobi para $(1, l + r, l + r + 1)$, con $(l + 1)/2 \leq r \leq (p - 1)/2 - l$. Esto nos da:

$$\alpha_{1,l+r}\alpha_{r+1,l+r+1} + \alpha_{l+r,l+r+1}\alpha_{l+2r+1,1} + \alpha_{l+r+1,1}\alpha_{r+2,l+r} = 0.$$

Si suponemos que $x_{l+r-1} = 0$, tenemos que

$$\alpha_{r+1,l+r+1} = \sum_{k=r+1}^{1+r+[(l-1)/2]} (-1)^{k-r-1}\binom{l-1}{k-r-1}x_k = 0$$

si $r \geq l$, porque $1 + r + [(l-1)/2] \leq r + l - 1$, y $\alpha_{1,l+r} = 0$. Por otro lado, tenemos que

$$\alpha_{r+2,l+r} = \sum_{k=r+2}^{r+[(l-1)/2]} (-1)^{k-r-2} \binom{l+r-1}{k-r-2} x_k = 0$$

si $r \geq l - 1$, mientras que $\alpha_{l+r+1,1} = 0$ si $r < l - 1$ $(1 + l + r + 1 \leq 2l)$. Tenemos también que

$$\alpha_{1,l+2r+1} = \sum_{k=1}^{1+r+[(l-1)/2]} (-1)^{k-i} \binom{l+2r-k}{k-1} x_k = (-1)^{k-l} \binom{2r-l}{l-1} x_l \neq 0,$$

de donde $x_{l+r} = 0$.

Por tanto, se sigue que $x_i = 0$ para $l + 1 \leq i \leq (p-1)/2$.

Recordemos que

$$0 = \alpha_{1,p} = \sum_{k=1}^{(p-1)/2} (-1)^{k-1} \binom{p-1-k}{k-1} = (-1)^{k-l} \binom{p-1-l}{l-1} x_l,$$

con lo que obtenemos la contradicción $x_l = 0$.

Para $l = 2$, argumentamos del mismo modo, teniendo en cuenta que, por la identidad de Jacobi aplicada a $(1, 3, 4)$,

$$\alpha_{1,3}\alpha_{2,4} + \alpha_{3,4}\alpha_{5,1} + \alpha_{4,1}\alpha_{3,3} = 0,$$

y, así, $\alpha_{3,4} = 0$, ya que $\alpha_{1,3} = \alpha_{3,3} = 0$ y $\alpha_{5,1} \neq 0$. □

Lema 4.4.2. *Supongamos que $l = 1$, $c_0 = p - 2$, entonces $2c \geq m - p - 1$. Más aún, las asignaciones $x_2 = \cdots = x_{(p-3)/2} = 0$ and $x_1 + x_2 = x_3 = \cdots = x_{(p-3)/2}$ satisfacen $\mathcal{S}(p)$.*

Demostración. Supongamos que $p+1 \leq m-2c-1$. En primer lugar, podemos aplicar Jacobi a la terna $(1, 2, 3)$, y obtenemos

$$\alpha_{1,2}\alpha_{p+1,3} + \alpha_{2,3}\alpha_{p+3,1} + \alpha_{3,1}\alpha_{p+2,2} = 0,$$

esto es, teniendo en cuenta la periodicidad módulo $p - 1$,

$$x_1x_2 + x_2\alpha_{4,1} - x_1x_2,$$

de donde $x_2(x_1 + x_2) = 0$. Por lo tanto, $x_2 = 0$ or $x_1 + x_2 = 0$.
Consideremos ahora Jacobi para la terna $(1, 2, 5)$. Tenemos:

$$x_1\alpha_{2,5} + \alpha_{2,5}\alpha_{6,1} + \alpha_{5,1}\alpha_{5,2} = 0. \tag{4.7}$$

Consideremos también Jacobi para $(1, 2, 6)$. Obtenemos

$$x_1\alpha_{2,6} + \alpha_{2,6}\alpha_{7,1} + \alpha_{6,1}\alpha_{6,2} = 0. \tag{4.8}$$

En el primer caso, si $x_2 = 0$ se sigue de (4.7) que

$$-x_1x_3 + x_3(x_1 + x_3) - x_1x_3 = 0,$$

de donde $x_3 = 0$ ó $x_3 - x_1 = 0$, y tenemos, por (4.8), que

$$-2x_1x_3 + 2x_3(x_1 + 3x_3) - 2(x_1 + x_3)x_3 = 0,$$

luego $x_3 = 0$ ó $4x_3 - 2x_1 = 0$. Por lo tanto, tenemos que $x_3 = 0$.
Supongamos que se da el segundo caso. Entonces (4.7) puede ser escrita en la forma

$$(-x_1 - x_3)(x_1 - 4x_1 - x_3 + 3x_1) = (-x_1 - x_3)x_3 = 0,$$

y (4.8) en la forma

$$(-x_1 - 2x_3)(x_1 - 5x_1 - 3x_3 + 4x_1 + x_3) = -2(-x_1 - 2x_3)x_3 = 0,$$

luego obtenemos también $x_3 = 0$.
Supongamos ahora, inductivamente, que $k \geq 4$, $2k + 1 \leq m - 2c - 1$ y $x_2 = x_3 = \cdots = x_{k-2} = 0$ o que $x_1 + x_2 = x_3 = \cdots = x_{k-2} = 0$. Consideremos Jacobi para $(1, 2, 2k - 2)$, esta ecuación puede ser escrita como

$$x_1\alpha_{2,2k-2} + \alpha_{2,2k-2}\alpha_{2k-1,1} + \alpha_{2k-2,1}\alpha_{2k-2,2} = 0,$$

esto es,

$$\alpha_{2,2k-2}(x_1 + \alpha_{2k-1,1} + \alpha_{2k-2,1}) = 0. \tag{4.9}$$

Consideremos también la identidad de Jacobi para $(1, 2, 2k - 3)$, obtenemos entonces que

$$x_1\alpha_{2,2k-3} + \alpha_{2,2k-3}\alpha_{2k-2,1} + \alpha_{2k-3,1}\alpha_{2k-3,2} = 0,$$

esto es,

$$\alpha_{2,2k-3}(x_1 + \alpha_{2k-2,1} + \alpha_{2,2k-3}) = 0. \tag{4.10}$$

En el primer caso, la hipótesis inductiva implica que $\alpha_{2,2k-3} = (-1)^{k-1}x_{k-1}$, $\alpha_{2,2k-2} = (-1)^{k-1}(k-2)x_{k-1}$, $\alpha_{1,2k-2} = x_1 + (-1)^k x_{k-1}$ and $\alpha_{1,2k-1} = x_1 + (-1)^k(k-1)x_{k-1}$, de donde (4.9) puede ser escrita como

$$(-1)^{k-1}(k-2)x_{k-1}\big(x_1 - x_1 - (-1)^k(k-1)x_{k-1} + x_1 + (-1)^k x_{k-1}\big) = 0,$$

esto es,

$$(-1)^{k-1}(k-2)x_{k-1}\big((-1)^k(-k+2)x_{k-1} + x_1\big) = 0,$$

y (4.10) puede escribirse como

$$(-1)^{k-1}x_{k-1}\big(x_1 - x_1 - (-1)^k x_{k-1} + x_1\big) = 0,$$

y si $x_{k-1} \neq 0$, obtenemos que

$$-(-1)^k x_{k-1} + x_1 = 0 = (-1)^k(-k+2)x_{k-1} + x_1,$$

luego

$$(-1)^k(-k+3)x_{k-1} = 0,$$

una contradicción.

En el segundo caso, la hipótesis inductiva implica que $\alpha_{2,2k-3} = -x_1 + (-1)^{k-1}x_{k-1}$, $\alpha_{2,2k-2} = -x_1 + (-1)^{k-1}(k-2)x_{k-1}$, $\alpha_{1,2k-2} = (2k-4)x_1 + (-1)^k x_{k-1}$ and $\alpha_{1,2k-1} = (2k-3)x_1 + (-1)^k(k-1)x_{k-1}$, luego las igualdades (4.9) y (4.10) pueden ser escritas como

$$\begin{aligned}&\big(-x_1 + (-1)^{k-1}(k-2)x_{k-1}\big)\\ &\cdot\big(x_1 - (2k-3)x_1 - (-1)^k(k-1)x_{k-1} + (2k-4)x_1 + (-1)^k x_{k-1}\big) = 0\\ &\big(-x_1 + (-1)^{k-1}x_{k-1}\big)\\ &\qquad\cdot\big(x_1 - (2k-4)x_1 - (-1)^k x_{k-1} + (2k-5)x_1\big) = 0, \end{aligned}\tag{4.11}$$

esto es,

$$\big(x_1 + (-1)^{k-1}(k-2)x_{k-1}\big)(-1)^k(-k+2)x_{k-1} = 0, \tag{4.12}$$

$$\big(-x_1 + (-1)^{k-1}x_{k-1}\big)\big(-(-1)^k x_{k-1}\big) = 0, \tag{4.13}$$

y si $x_{k-1} \neq 0$, entonces obtenemos que

$$-x_1 + (-1)^{k-1}(k-2)x_{k-1} = 0 = -x_1 + (-1)^{k-1}x_{k-1},$$

luego $(-1)^{k-1}(k-1)x_{k-1} = 0$ y, por consiguiente, $x_{k-1} = 0$ en este segundo caso. □

4.5. Cotas del tipo $2c \geq m - 2l - c_0 - 1$

Lema 4.5.1. *Si $3l + 2 \leq m - 2c - 1$ y $l \geq 2$, entonces se da una de las siguientes condiciones:*

1. $\alpha_{1,2l+c_0+l} \neq 0$, $\alpha_{1,2l+k} = \binom{l+k-1}{l-1}\alpha_{1,2l}$, $x_{l+k} = 0$ *para* $1 \leq k \leq \left[\frac{l}{2}\right]$.
2. $\alpha_{1,2l+c_0+l} = \alpha_{1,2l+c_0+l+1} = 0$.
3. $\alpha_{1,2l+c_0+l} = 0$, $\alpha_{1,2l+c_0+l+1} \neq 0$, $x_{l+1} + x_l = 0$, $x_{l+k} = 0$ *para* $2 \leq k \leq \left[\frac{l}{2}\right]$.

Demostración. Tenemos dos posibilidades:

1. Si $\alpha_{1,2l+c_0+l} = z \neq 0$, aplicamos la identidad de Jacobi a las ternas $(l-k, l, l+k+1)$ para $1 \leq k \leq l-1$ para obtener

$$\begin{aligned} 0 &= \alpha_{l,l+k+1}(-1)^{l-k}z + (-1)^{l-k+1}\alpha_{1,2l}(-1)^{l-1}z \\ &= (-1)^{l-k}z\big(\alpha_{l,l+k-1} + (-1)^l\alpha_{1,2l}\big), \end{aligned}$$

   luego

$$\alpha_{l,l+k+1} = (-1)^{l-1}\alpha_{1,2l}, \qquad 1 \leq k \leq l-1.$$

   Por tanto, como los valores de la columna l-ésima columna son iguales, obtenemos que $x_{l+k} = 0$ para $1 \leq k \leq \left[\frac{l+1}{2}\right] - 1$. Pero, entonces,

$$\begin{aligned} \alpha_{1,2l+k} &= \sum_{\lambda=1}^{\left[\frac{2l+k}{2}\right]} \binom{2l+k-\lambda-1}{\lambda-1}(-1)^{\lambda-1}x_\lambda \\ &= \binom{l+k-1}{l-1}\alpha_{1,2l}, \qquad 1 \leq k \leq l-1, \end{aligned}$$

   luego se tiene el primer caso.

2. Si $\alpha_{1,2l+c_0+l} = 0$, consideramos el valor de $\alpha_{1,2l+c_0+l+1}$. Este valor puede ser igual o distinto de cero. Si $\alpha_{1,3l+c_0+1} = 0$, se da el segundo caso. Si $\alpha_{1,3l+c_0+1} = z \neq 0$, podemos aplicar la identidad de Jacobi a las ternas $(l-k, l, l+k+2)$, donde $1 \leq k \leq l-1$, y deducimos que

$$\begin{aligned} 0 &= \alpha_{l,l+k+2}(-1)^{l-k}z + (-1)^{l-k-1}(k-1)\alpha_{1,2l}(-1)^{l-1}z \\ &= (-1)^{l-k}z\big(\alpha_{l,l+k+2} + (-1)^l(k+1)\alpha_{1,2l}\big), \end{aligned}$$

luego $\alpha_{l,l+k+2} = (-1)^{l-1}(k+1)\alpha_{1,2l}$. Pero entonces

$$x_{l+1} = \alpha_{l+1,l+2} = \alpha_{l,l+2} - \alpha_{l,l+3} = (-1)^{l-1}(l-2)\alpha_{1,2l} = -\alpha_{l,l+1} = -x_l$$

y

$$\begin{aligned}\alpha_{l+1,l+2+k} &= \alpha_{l,l+2+k} - \alpha_{l,l+3+k} \\ &= (-1)^{l-1}\big(k-(k+1)\big)\alpha_{1,2l} \\ &= -\alpha_{l,l+1} \\ &= -x_l,\end{aligned}$$

luego tenemos el mismo valor a lo largo de toda la $l+1$-ésima columna, y esto implica que $x_{l+k} = 0$ para $2 \leq k \leq \left[\frac{l+1}{2}\right] - 1$.

□

Lema 4.5.2. *Supongamos que* $2l+c_0+1 \leq m-2c-1$, $c_0 \geq l \geq 3$, p *no divide* $c_0 + 2l$ *if* c_0 *y* l *son ambos impares, y* $c_0 < (p-l)/2$, *entonces* $\alpha_{1,2l+c_0+i} = 0$ *para* $1 \leq i \leq l$.

Demostración. Por reducción al absurdo, supongamos, que $\alpha_{1,3l+c_0} \neq 0$. Expresemos $2l+c_0+1 = 3l+2\mu+e$, con $e \in \{0,1\}$ y $\mu \geq 1$. Como $3l \leq m-2c-1$, por el Lema 4.2.1 obtenemos que $\alpha_{1,2l+c_0+k} = 0$ para $1 \leq k \leq l-1$. Si $\alpha_{1,3l+c_0} = z \neq 0$, podemos aplicar la identidad de Jacobi a las ternas $(l-k, l, l+k+1)$ para $1 \leq k \leq l-1$ para obtener

$$\begin{aligned}0 &= \alpha_{l,l+k+1}(-1)^{l-k}z + (-1)^{l-k+1}\alpha_{1,2l}(-1)^{l-1}z \\ &= (-1)^{l-k}z\big(\alpha_{l,l+k+1} + (-1)^l\alpha_{1,2l}\big),\end{aligned}$$

luego

$$\alpha_{l,l+k+1} = (-1)^{l-1}\alpha_{1,2l}, \qquad 1 \leq k \leq l-1.$$

Por tanto, deducimos que $x_{l+k} = 0$ para $1 \leq k \leq \left[\frac{l+1}{2}\right] - 1$. En consecuencia, tenemos que

$$\begin{aligned}\sum_{\lambda=1}^{\left[\frac{2l+k}{2}\right]} \binom{2l+k-\lambda-1}{\lambda-1}(-1)^{\lambda-1}x_\lambda & \\ &= \binom{l+k-1}{l-1}(-1)^{l-1}x_l \\ &= \binom{l+k-1}{l-1}\alpha_{1,2l}, \qquad 1 \leq k \leq l-1.\end{aligned}$$

En particular, todos los valores de la submatriz T_{3l} vienen dados por la fórmula

$$\alpha_{l-i,l+j} = \binom{j-1}{i}(-1)^{l+i-1}\alpha_{1,2l}, \qquad 1 \le i \le l-1, 1 \le j \le l+i. \quad (4.14)$$

Escribamos $l = 2l' + f$, con $f \in \{0, 1\}$. Afirmamos que se tiene la siguiente proposición:

> Si $3l+2k \le m-2c-1$, $2k \le p-l$ y $k \le \mu$, entonces $\alpha_{l+1,2l+c_0+i} = 0$ para $1 \le i \le 2k-1$ y $x_{l+i} = 0$ para $1 \le i \le [l/2]+k-1$. Además, if $3l+2k+1 \le m-2c-1$, también tenemos que $\alpha_{l+1,2l+c_0+2k} = 0$.

Probamos esta afirmación por inducción sobre k.

Si $k = 1$, tenemos que $3l+2 \le m-2c-1$. Si aplicamos la identidad de Jacobi a la terna $(l-1, l+1, l+2)$, tenemos que $0 = x_l\alpha_{l+1,2l+1+c_0}$, luego $\alpha_{l+1,2l+1+c_0} = 0$. La terna $(1, l+[l/2], l+[l/2]+1)$ origina $0 = \alpha_{l+[l/2],l+[l/2]+1}\alpha_{3l+c_0+1-f,1} = 0$, de donde $x_{l+[l/2]}\alpha_{1,3l+c_0+1-f} = 0$ y, por tanto, $x_{l+[l/2]} = 0$. Finalmente, supongamos que $3l+3 \le m-2c-1$, entonces podemos considerar la terna $(l, l+1, l+2)$, que origina

$$\begin{aligned} 0 &= x_l(\alpha_{l+1,2l+2+c_0} - \alpha_{l+2,2l+1+c_0}) \\ &= x_l\big(w - (-w)\big) \\ &= 2x_l w, \end{aligned}$$

siendo $w = \alpha_{l+1,2l+2+c_0}$, luego $\alpha_{l+1,2l+2+c_0} = 0$.

Supongamos que nuestra afirmación es válida para $k \le \lambda$, demostrémosla para $\lambda+1$ con $2(\lambda+1) \le p-l$. Tenemos entonces que $3l+2(\lambda+1) \le m-2c-1$, en particular, $3l+2\lambda+1 \le m-2c-1$ y, por la hipótesis inductiva,

$$\alpha_{l+1,2l+c_0+i} = 0, \qquad 1 \le i \le 2\lambda$$

y

$$x_{l+i} = 0, \qquad 1 \le i \le [l/2]+\lambda-1.$$

Tenemos también que

$$\alpha_{1,2l+c_0+l+i} = \binom{l+i-1}{i}z, \qquad 0 \le i \le 2\lambda.$$

Sólo nos resta probar que $\alpha_{l+1,2l+c_0+2\lambda+1} = 0$ y $x_{l+[l/2]+\lambda} = 0$. En efecto, la terna $(l-1, l+\lambda+1, l+\lambda+2)$ implica

$$\begin{aligned} 0 &= -\lambda x_l\big(-(-1)^{\lambda+1}\big)w - (\lambda+1)x_l(-1)^{\lambda}w \\ &= (-1)^{\lambda}(2\lambda+1)x_l w, \end{aligned}$$

con $w = \alpha_{l+1,2l+2\lambda+1+c_0}$, ya que $x_{l+\lambda+1} = 0$. Por tanto,

$$\alpha_{l+1,2l+2\lambda+1+c_0} = 0,$$

y deducimos que

$$\alpha_{1,2l+c_0+l+1} = \binom{l+i-1}{i} z, \qquad i = 2\lambda+1.$$

La terna$(1, l+[l/2]+\lambda, l+[l/2]+\lambda+1)$ implica

$$\begin{aligned} 0 = {} & \alpha_{1,l+[l/2]+\lambda}\alpha_{l+[l/2]+\lambda+c_0+1,l+[l/2]+\lambda+1} \\ & + \alpha_{l+[l/2]+\lambda,l+[l/2]+\lambda+1}\alpha_{3l+2\lambda+c_0+1-f,1} \\ & + \alpha_{l+[l/2]+\lambda+1,1}\alpha_{l+[l/2]+\lambda+2+c_0,l+[l/2]+\lambda}. \end{aligned} \tag{4.15}$$

Pero:

1. Si $\lambda < (l/2)+1$, entonces $\alpha_{1,l+[l/2]} = \alpha_{1,l+[l/2]+\lambda+1} = 0$ y de (4.15) deducimos que
$$0 = x_{l+[l/2]+\lambda}\alpha_{3l+2\lambda+c_0+1-f,1}.$$

2. Si l es par y $\lambda = (l/2)-1$, entonces $\alpha_{1,l+(l/2)+\lambda} = 0$ y, por la hipótesis de inducción, $\alpha_{l+(l/2)+\lambda,l+(l/2)+\lambda+2+c_0} = \alpha_{2l-1,2l+1+c_0} = 0$, luego (4.15) implica
$$0 = x_{l+(l/2)+\lambda}\alpha_{3l+2\lambda+c_0+1,1} = 0.$$

3. Si l es par y $\lambda = [l/2] = (l-1)/2$ entonces $\alpha_{1,l+[l/2]+[l/2]} = 0$ y
$$\alpha_{l+[l/2]+[l/2],l+[l/2]+[l/2]+2+c_0} = \alpha_{2l-1,2l+1+c_0} = 0$$
por la hipótesis de inducción, luego (4.15) implica que
$$0 = x_{l+[l/2]+\lambda}\alpha_{3l+2\lambda+c_0,1} = 0.$$

4. Si $\lambda \geq l/2$, entonces $l+1 < l+[l/2]+\lambda+1 \leq l+2\lambda+1$ y

$$\alpha_{l+[l/2]+\lambda,l+[l/2]+\lambda+2+c_0} = \alpha_{l+[l/2]+\lambda+1,l+[l/2]+\lambda+1+c_0} = 0,$$

luego (4.15) puede expresarse como

$$0 = x_{l+[l/2]+\lambda}\alpha_{3l+2\lambda+c_0+1-f,1}.$$

Luego, en cualquier caso, (4.15) es equivalente a

$$0 = x_{l+[l/2]+\lambda}\alpha_{3l+2\lambda+c_0+1-f,1} = -x_{l+[l/2]+\lambda}\binom{l+2\lambda-f}{2\lambda-1-f}z.$$

Como $2\lambda+l-f < p-l+l = p$, tenemos que

$$x_{l+[l/2]+\lambda} = 0.$$

Para concluir, si $3l+2(\lambda+1)+1 \leq m-2c-1$, entonces, teniendo en cuenta que $x_{l+\lambda+1} = 0$, la terna $(l, l+\lambda+1, l+\lambda+2)$ implica

$$0 = x_l\big(-(-1)^{\lambda+1}\big)w + x_l(-1)^{\lambda}w = (-1)^{\lambda}2x_l w,$$

siendo $w = \alpha_{l+1,2l+2\lambda+1+c_0}$.
En consecuencia, nuestra afirmación es válida para $k < (p-l)/2$ y los valores de T_{3l+2k} vienen dados por

$$\alpha_{l-i,l+j} = \binom{j-1}{i}(-1)^{l+i-1}y_0, \qquad 1 \leq i \leq l-1, \quad 1 \leq j \leq l+i.$$

Entonces, como $c_0 < (p-l)/2$, tomando $k = \mu$ tenemos determinada la matriz $T_{2l+c_0+2} = T_{3l+2\mu+e+1}$.
De este modo, sabemos que

$$\begin{aligned} 0 &= \alpha_{1,2l+c_0+1} \\ &= \sum_{\lambda=1}^{[(2l+c_0+2-1)/2]} (-1)^{\lambda-1}\binom{2l+c_0+1-\lambda-1}{\lambda-1}x_\lambda \end{aligned}$$

Además, $x_\lambda = 0$ si $l+1 \leq \lambda \leq l + [l/2] + \mu - 1 = l + \frac{c_0+1-f-e}{2}$, luego si l es par o c_0 es par, $l + [l/2] + \mu - 1 = [(2l + c_0 + 1)/2]$ y

$$0 = \alpha_{1,2l+c_0+1} = (-1)^{l-1}\binom{l+c_0}{l-1}x_l,$$

una contradicción, ya que $l+c_0 < (p+l)/2 < p$, y si l y c_0 son ambos impares,

$$\begin{aligned} 0 &= \alpha_{1,2l+c_0+1} \\ &= (-1)^{l-1}\binom{l+c_0}{l-1}x_l + (-1)^{l+(c_0-1)/2}\binom{l+(c_0-1)/2}{l+(c_0-1)/2}x_{l+(c_0+1)/2}, \end{aligned}$$

mientras que

$$\begin{aligned} 0 &= \alpha_{1,2l+c_0+2} \\ &= (-1)^{l-1}\binom{l+c_0+1}{l-1}x_l + (-1)^{l+(c_0-1)/2}\binom{l+(c_0+1)/2}{l+(c_0-1)/2}x_{l+(c_0+1)/2}, \end{aligned}$$

por tanto

$$\Big(\big(l+(c_0+1)/2\big)\binom{l+c_0}{l-1} - \binom{l+c_0+1}{l-1}\Big)x_l = 0,$$

esto es,

$$\big(l+(c_0+1)/2\big)(c_0+2) - (l+c_0+1) \equiv 0 \pmod{p},$$

en consecuencia p divide $(c_0+1)(c_0+2l)/2$, una contradicción. □

Denotemos $y_j = \alpha_{1,2l+j}$.

Lema 4.5.3. *Si $3l + \mu \leq m - 2c - 1$ y $l \geq \mu + 2 \geq 2$, entonces se tiene uno de los siguientes casos:*

1. $y_{c_0+l+i} = 0$ *para* $0 \leq i \leq \mu - 1$.
2. $y_{c_0+l} \neq 0$, $y_i = \binom{l+i-1}{l-1}y_0$, $x_{l+i} = 0$ *para* $1 \leq i \leq [l/2]$.
3. $y_{c_0+l+i} = 0$ *para* $0 \leq i \leq \mu - 2$, $y_{c_0+l+\mu-1} \neq 0$, $x_{l+\mu-1} + (-1)^\mu x_l = 0$, $x_{l+\mu-2} - (-1)^\mu(\mu-1)x_l = 0$ *y* $x_{l+i} = 0$ *para* $\mu \leq i \leq [(l+\mu-2)/2]$.

Además, en cualquier caso $y_{c_0+i} = 0$ *para* $1 \leq i \leq l-1$.

Demostración. Como $3l \leq 3l+\mu \leq m-2c-1$, de 4.2.1 deducimos que $y_{c_0+i} = 0$ para $1 \leq i \leq l-1$.

De cara a probar la otra afirmación, argumentamos por inducción. Para $\mu = 2$, la afirmación se tiene por el Lema 4.5.1. Supongamos que es cierta para $\lambda < \mu$, y probémosla para $\lambda + 1 \leq \mu$. Como $3l+\lambda \leq 3l+\lambda+1 \leq 3l+\mu \leq m-2c-1$, por la hipótesis inductiva deducimos que se da una de las siguientes situaciones:

1. $y_{c_0+l+i} = 0$ para $0 \leq i \leq \lambda - 1$.

2. $y_{c_0+l} \neq 0$, $y_k = \binom{l+k-1}{l-1} y_0$, $x_{l+k} = 0$ para $1 \leq k \leq [l/2]$.

3. $y_{c_0+l+i} = 0$ para $0 \leq i \leq \lambda - 2$, $y_{c_0+l+\lambda-1} \neq 0$, $x_{l+\lambda-1} + (-1)^\lambda x_l = 0$, $x_{l+\lambda-2} - (-1)^\lambda(\lambda-1)x_l = 0$ y $x_{l+i} = 0$ para $\lambda \leq i \leq [(l+\lambda-2)/2]$.

Afirmamos que si $y_{c_0+l} = 0$ y $3l+\lambda+1 \leq m-2c-1$, entonces $y_{c_0+l+\lambda-1} = 0$, o sea, no se da la tercera posibilidad. Efectivamente, si $y_{c_0+l+\lambda-1} = z \neq 0$, se da la tercera posibilidad, y usando la terna $(l-\lambda+2, l+\lambda-1, l+\lambda)$ tenemos:

$$\begin{aligned}
0 &= \alpha_{l-\lambda+2,l+\lambda-1}\alpha_{2l+1+c_0,l+\lambda} + \alpha_{l+\lambda-1,l+\lambda}\alpha_{2l+\lambda-1+c_0,l-\lambda+2} \\
&\quad + \alpha_{l+\lambda,l-\lambda+2}\alpha_{2l+2+c_0,l+\lambda-1} \\
&= (-1)^{l-\lambda+1} y_0 (-1)^{l+\lambda+1}\big((l+\lambda-1)z - w_1\big) \\
&\quad - x_{l+\lambda-1}(-1)^{l-\lambda+2}\big((l-\lambda+2-1)z - w_1\big) \\
&\quad + (-1)^{l-\lambda+1}\big(l-(l-\lambda+2)+1\big) y_0 (-1)^{l+\lambda-1}\big((l+\lambda-2)z - w_1\big) \\
&= (-1)^{l+1} x_l \big((l+\lambda-1)z - w_1\big) \\
&\quad + (-1)^{l+2} x_l \big((l-\lambda+1)z - w_1\big) \\
&\quad + (-1)^{l-1} x_l (\lambda-1)\big((l+\lambda-2)z - w_1\big) \\
&= (-1)^{l-1}(\lambda-1) x_l \big((l+\lambda)z - w_1\big),
\end{aligned}$$

con $w_1 = \alpha_{1,3l+\lambda+c_0}$, de donde $w_1 = (l+\lambda)z$. Pero si consideramos la terna $(l-\lambda, l+\lambda, l+\lambda+1)$, teniendo en cuenta que $x_{l+\lambda} = 0$ por la hipótesis de

inducción se tiene que

$$\begin{aligned}
0 &= \alpha_{l-\lambda,l+\lambda}\alpha_{2l+c_0,l+\lambda+1} + \alpha_{l+\lambda,l+\lambda+1}\alpha_{2l+\lambda+1+c_0,l-\lambda} \\
&\quad + \alpha_{l+\lambda+1,l-\lambda}\alpha_{2l+1+c_0,l+\lambda} \\
&= (-1)^{l-\lambda-1}y_0\big((-1)^{l+\lambda}(l+\lambda-1)z + (-1)^{l+\lambda-1}(l+\lambda)z\big) \\
&= (-1)^{l-\lambda-1}y_0(-1)^{l+\lambda}\big((l+\lambda-1)z - (l+\lambda)z\big) \\
&= (-y_0)(-z) \\
&= y_0 z,
\end{aligned}$$

por tanto $y_0 = 0$ ó $z = 0$, lo cual es imposible. En consecuencia, si $y_{c_0+l} = 0$ y $3l + \lambda + 1 \leq m - 2c - 1$, entonces $y_{c_0+l+\lambda-1} = 0$.

Si $y_{c_0+l+\lambda} = 0$, tenemos el primer caso del Lema para $\lambda+1$. Si $z = y_{c_0+l+\lambda} \neq 0$ hemos de probar las condiciones dadas en el tercer caso del Lema para $\lambda+1$. En efecto, si tomamos las ternas $(l-k, l+\lambda, l+k+1)$, con $\lambda \leq k \leq l-1$, deducimos que

$$\begin{aligned}
0 &= \alpha_{l-k,l+\lambda}\alpha_{2l-k+\lambda+c_0,l+k+1} + \alpha_{l+\lambda,l+k+1}\alpha_{2l+\lambda+k+1+c_0,l-k} \\
&\quad + \alpha_{l+k+1,l-k}\alpha_{2l+1+c_0,l+\lambda} \\
&= (-1)^{l-k-1}z\big(-\alpha_{l+\lambda,l+k+1} + (-1)^{l+\lambda-1}y_0\big),
\end{aligned}$$

y así $\alpha_{l+\lambda,l+k+1} = (-1)^{l+\lambda-1}y_0 = (-1)^{\lambda}x_l$. En consecuencia, tenemos que $x_{l+\lambda} = (-1)^{\lambda}x_l$ y $x_{l+i} = 0$ para $\lambda + 1 \leq i \leq [(l+\lambda-1)/2]$.

Además, si usamos la terna $(l-\lambda+2, l+\lambda-1, l+\lambda+1)$, obtenemos que $z(x_{l+\lambda-1} - (-1)^{\lambda+1}\lambda x_l) = 0$, por tanto $x_{l+\lambda-1} - (-1)^{\lambda+1}\lambda x_l = 0$ y se verifica el Lema para $\lambda + 1$. Esto completa la demostración. □

Lema 4.5.4. *Supongamos que* $2l + c_0 + 1 \leq m - 2c - 1$, $l \geq (p+5)/6$ *y* $l + c_0 = (p-1)/2$. *Entonces* $\alpha_{i,2l+c_0} = 0$ *para* $1 \leq i \leq c_0$.

Demostración. Expresemos $2l + c_0 + 1 = 3l + \mu$, con $\mu \geq 0$ (recordemos que $c_0 - l + 1 = (p-1)/2 - 2l + 1 \geq 0$). Además, $l \geq \mu + 2$ si, y sólo si, $l \geq c_0 + 3 - l$, esto es, $2l \geq c_0 + 3$, pero $c_0 = (p-1)/2 - l$, por tanto, $2l \geq (p-1)/2 - l + 3$ si, y sólo si, $3l \geq (p-1)/2 + 3 = (p+5)/2$, o sea, $l \geq (p+5)/6$, la condición sobre la hipótesis.

Como $l \geq \mu + 2$ y $3l + \mu \leq m - 2c - 1$, tenemos, por el Lema 4.5.3, que se verifica una de las siguientes condiciones:

1. $y_{c_0+l+i} = 0$ for $0 \leq i \leq \mu - 1$.

2. $y_{c_0+l} \neq 0$, $y_k = \binom{l+k-1}{l-1} y_0$, $x_{l+k} = 0$ para $1 \leq k \leq [l/2]$.

3. $y_{c_0+l+i} = 0$ para $0 \leq i \leq \mu - 2$, $y_{c_0+l+\mu-1} \neq 0$, $x_{l+\mu-1} + (-1)^\mu x_l = 0$, $x_{l+\mu-2} - (-1)^\mu(\mu-1)x_l = 0$, $x_{l+i} = 0$ para $\mu \leq i \leq [(l+\mu-2)/2]$.

Además, $3l \leq m - 2c - 1$, luego $y_{c_0+1} = \cdots = y_{c_0+l-1} = 0$ y $2l + c_0 + 1 \leq m - 2c - 1$, $c_0 \geq l \geq 3$ y $c_0 \leq (p-l)/2$ si, y sólo si, $2c_0 + l \leq p$ o, equivalentemente, $2c_0 + l \leq p$, esto es, usando la condición $c_0 + l = (p-1)/2$, $2\big((p-1)/2 - l\big) + l \leq p$, una condición equivalente a $p - 1 - 2l + l \leq p$, que siempre se tiene, ya que $-1 - l \leq 0$. Luego $y_{c_0+l} = 0$ y no se puede dar el segundo caso. Luego se da la primera o la tercera condición. En cualquier caso, se tiene que $y_{c_0+\delta} = 0$ para $1 \leq \delta \leq \mu - 2 + l$, pero $l + \mu = c_0 + 1$. Por tanto,

$$y_{c_0+\delta} = 0 \qquad \text{para } 1 \leq \delta \leq c_0 - 1.$$

□

Lema 4.5.5. *Supongamos que $l + c_0 = (p-1)/2$ y $l \geq (p+5)/6$. Entonces $2c \geq m - 2l - c_0 - 1$ ó $c = c_0 = (p-1)/2 - l$.*

Demostración. Supongamos que $2l + c_0 + 1 \leq m - 2c - 1$.

Por 4.5.4, tenemos que $\alpha_{1,2l+c_0+1} = \cdots = \alpha_{1,2l+c_0+c_0-1} = 0$. Llamemos

$$\alpha_{1,2l+2c_0} = \alpha_{1,p-1} = z,$$

tenemos que $\alpha_{1,2l+2c_0+i} = \alpha_{1,p-1+i} = \alpha_{1,i} = 0$ para $1 \leq i \leq 2l-1$. Obtenemos que $2l + c_0 + 1 \leq m - 2c - 1$, luego $2l + c_0 + 1 + c \leq m - c - 1$. Hay dos posibilidades: o $c = c_0 = (p-1)/2 - l$ o $c > c_0$, luego $c \geq c_0 + p - 1$ y $2l + 2c_0 + p \leq m - c - 1$. Supongamos que se da el último caso, obtenemos una columna de ceros (para todos los valores excepto $\alpha_{1,2l+2c_0} = \alpha_{1,p-1} = z$) desde $\alpha_{1,2l+c_0+1}$ hasta $\alpha_{1,2l+2c_0+2l-1}$, esto es, una columna de longitud $c_0 + 2l - 1$, que crea un triángulo con lados de longitud $c_0 + 2l - 1$, esto es, $c_0 + 2l - 1$ columnas. El peldaño extremo de la fila $i + j = 1 + 2l + 2c_0 + 2l - 1 = 4l + 2c_0$ corresponde a $i + i + 2 = 4l + 2c_0$, por tanto $i = 2l + c_0 - 1$ y, de este modo, el triángulo $T_{2l+2c_0+2l} = T_{p-1+2l}$ puede darse en términos de z, en particular, por la definición de l, $z \neq 0$.

Tenemos una fórmula que nos da los valores del triángulo en función de su primera columna,

$$\alpha_{i,j} = \sum_{w=j}^{j+i-1} (-1)^{w-j} \binom{i-1}{w-j} \alpha_{1,w}.$$

Los valores $\alpha_{1,2l+c_0}$, $\alpha_{1,2l+c_0-1}$, $\alpha_{1,2l+c_0-2}$ pueden ser dados en términos de z. Tenemos:

$$\begin{aligned}
\alpha_{c_0+2l-1,c_0+2l+1} &= (-1)^{c_0-1}\binom{c_0+2l-2}{c_0-1}z,\\
\alpha_{c_0+2l-1,c_0+2l} &= (-1)^{c_0}\binom{c_0+2l-2}{c_0}z+\alpha_{1,c_0+2l},\\
\alpha_{c_0+2l-2,c_0+2l} &= (-1)^{c_0}\binom{c_0+2l-3}{c_0}z+\alpha_{1,c_0+2l},\\
\alpha_{c_0+2l-2,c_0+2l-1} &= (-1)^{c_0+1}\binom{c_0+2l-3}{c_0+1}z+\alpha_{1,c_0+2l-1}\\
&\quad -\binom{c_0+2l-3}{1}\alpha_{1,c_0+2l},\\
\alpha_{c_0+2l-3,c_0+2l-1} &= (-1)^{c_0+1}\binom{c_0+2l-4}{c_0+1}z+\alpha_{1,c_0+2l-1}\\
&\quad -\binom{c_0+2l-4}{1}\alpha_{1,c_0+2l},\\
\alpha_{c_0+2l-3,c_0+2l-2} &= (-1)^{c_0+2}\binom{c_0+2l-4}{c_0+2}z+\alpha_{1,c_0+2l-2}\\
&\quad -(c_0+2l-4)\alpha_{1,c_0+2l-1}+\binom{c_0+2l-4}{2}\alpha_{1,c_0+2l}.
\end{aligned}$$

Por tanto, teniendo en cuenta que

$$\begin{aligned}
\alpha_{c_0+2l-1,c_0+2l+1} &= \alpha_{c_0+2l-1,c_0+2l},\\
\alpha_{c_0+2l-2,c_0+2l} &= \alpha_{c_0+2l-2,c_0+2l-1},
\end{aligned}$$

y

$$\alpha_{c_0+2l-3,c_0+2l-1} = \alpha_{c_0+2l-3,c_0+2l-2},$$

obtenemos que

$$\begin{aligned}
\alpha_{1,c_0+2l} &= (-1)^{c_0+1}\binom{c_0+2l-1}{c_0}z,\\
\alpha_{1,c_0+2l-1} &= \binom{c_0+2l-2}{1}\alpha_{1,c_0+2l} + (-1)^{c_0}\binom{c_0+2l-2}{c_0+1}z,\\
\alpha_{1,c_0+2l-2} &= \binom{c_0+2l-3}{1}\alpha_{1,c_0+2l-1} - \binom{c_0+2l-3}{2}\alpha_{1,c_0+2l}\\
&\quad + (-1)^{c_0+1}\binom{c_0+2l-3}{c_0+2}z.
\end{aligned}$$

Por otro lado, si consideramos las ternas $(1, i, c_0+2l-i)$ para $2 \le i \le 2l-1$, tenemos que

$$\alpha_{i,c_0+2l-i}(-z) + \alpha_{1,c_0+2l-i}(-1)^{i-1}z = 0,$$

luego

$$\alpha_{i,c_0+2l-i} = (-1)^{i-1}\alpha_{1,c_0+2l-i}, \qquad 2 \le i \le 2l-1.$$

En particular, $\alpha_{2,c_0+2l-2} = -\alpha_{1,c_0+2l-1}$, pero

$$\alpha_{1,c_0+2l-1} + \alpha_{2,c_0+2l-2} = \alpha_{1,c_0+2l-2},$$

esto es,

$$2\alpha_{1,c_0+2l-2} = \alpha_{1,c_0+2l-1}.$$

Podemos substituir los valores calculados en esta ecuación, con lo que obtenemos

$$\begin{aligned}
2\Bigg(&\binom{c_0+2l-3}{1}\alpha_{1,c_0+2l-1}\\
&\quad - \binom{c_0+2l-3}{2}\alpha_{1,c_0+2l} + (-1)^{c_0+1}\binom{c_0+2l-3}{c_0+2}z\Bigg)\\
&\qquad = \binom{c_0+2l-2}{1}\alpha_{1,c_0+2l} + (-1)^{c_0}\binom{c_0+2l-2}{c_0+1}z,
\end{aligned}$$

y, tras algunas substituciones y cálculos, y tras simplificar $(-1)^{c_0}z$, llegamos

a la ecuación

$$\left(-2(c_0+2l-3)(c_0+2l-2)+2\binom{c_0+2l-3}{2}+(c_0+2l-2)\right)\binom{c_0+2l-1}{c_0}$$
$$+(2(c_0+2l-3)-1)\binom{c_0+2l-2}{c_0+1}-2\binom{c_0+2l-3}{c_0+2}\equiv 0 \pmod{p}.$$

Llamemos $x = c_0 + 2l$, simplificando los números combinatorios, obtenemos la ecuación

$$\begin{aligned}&(c_0+2)(c_0+1)(x-1)(x-2)(-x^2+4x-2)\\ &\quad+(2x-7)(x-2)(x-c_0-1)(x-c_0-1)(x-c_0-2)(c_0+2)\\ &\quad-2(x-c_0-1)(x-c_0-2)(x-c_0-3)(x-c_0-4)\equiv 0 \pmod{p}\end{aligned}$$

Como $x = c_0+2l \equiv -c_0-1 \pmod{p}$ (ya que $2c_0+2l+1 = p-1+1 = p \equiv 0 \pmod{p}$, podemos transformar nuestra ecuación en

$$\begin{aligned}&(c_0+2)(c_0+1)\big(-(c_0+2)\big)\big(-(c_0+3)\big)\big(-(c_0+1)^2-4(c_0+1)-2\big)\\ &\quad+\big(-2(c_0+1)-7\big)(-1)(c_0+3)(-1)(2c_0+2)(2c_0+3)(-1)(c_0+2)\\ &-2\big(2(c_0+1)\big)(-1)(-1)(2c_0+3)(-1)(2c_0+4)(-1)(2c_0+5)\equiv 0 \pmod{p},\end{aligned}$$

y, tras más cálculos, llegamos a

$$-(c_0^4+3c_0^3+3c_0^2+c_0) = -c_0(c_0+1)^3 \equiv 0 \pmod{p},$$

una contradicción. □

4.6. Cotas del tipo $2c \geq m - p - 2l + c_0 - 1$

Lema 4.6.1. *Supongamos que* $p + 2l - c_0 + 1 \leq m - 2c - 1$. *Entonces* $\alpha_{j,p-c_0+1} = 0$ *para* $1 \leq j \leq l-2$, $l+1 \leq j \leq 2l-1$, *y* $\alpha_{l-1,p-c_0+1}+\alpha_{l,p-c_0+1} = 0$.

Demostración. En el Lema 2.6.1 se ha probado que $z_j = z_1$ para todo $j \in [1, 2l] - \{l\}$, donde $z_j = \alpha_{j,p-c_0}$. Considerando la diagonal $(p - c_0 + 1)$, deducimos que

$$\alpha_{j,p-c_0+1} = \alpha_{j,p-c_0} - \alpha_{j+1,p-c_0} = 0 \quad \text{para } j \in [1, l-2] \cup [l+1, 2l-1],$$

y

$$\alpha_{l-1,p-c_0+1} = \alpha_{l-1,p-c_0} - \alpha_{l,p-c_0} = z_1 - z_l,$$

$$\alpha_{l,p-c_0+1} = \alpha_{l,p-c_0} - \alpha_{l+1,p-c_0} = z_l - z_1,$$

de donde, sumando, obtenemos que $\alpha_{l-1,p-c_0+1} + \alpha_{l,p-c_0+1} = 0$. □

Supongamos que $l \geq 3$. Tenemos, para $j \in [1, l-2] \cup [l+1, 2l-1]$,

$$0 = \alpha_{j,p-c_0+1} = \sum_{\lambda=i}^{[(j+p-c_0)/2]} (-1)^{\lambda-j} \binom{p-c_0+1-\lambda-1}{\lambda-j} x_\lambda, \tag{4.16}$$

bajo la hipótesis $p + 2l - c_0 + 1 \leq m - 2c - 1$. Las variables x_λ que aparecen aquí son

$$x_l, x_{l+1}, \ldots, x_{[(2l-1+p-c_0+1-1)/2]} = x_{l+[(-1+p-c_0)/2]},$$

esto es, hay

$$[(-1+p-c_0)/2] + 1$$

variables y el número de ecuaciones que origina el primer Lema es $2l-1-2 = 2l-3$ más las correspondiente a $\alpha_{l-1,p-c_0+1} + \alpha_{l,p-c_0+1} = 0$, esto es, tenemos un sistema de $2l-2$ ecuaciones con $[(-1+p-c_0)/2] + 1$ incógnitas.

Para $p - c_0 = 4l - 4$, tenemos $[(-1+p-c_0)/2] + 1 = 2l - 2$, considerando las $2l-2$ ecuaciones dadas en el Lema anterior.

Para $p - c_0 = 4l - 6$, tenemos $[(-1+p-c_0)/2] + 1 = 2l - 3$ incógnitas, y consideramos las primeras $2l-3$ ecuaciones.

Para $p - c_0 = 4l - 8$, tenemos $[(-1+p-c_0)/2] + 1 = 2l - 4$ incógnitas y consideramos las primeras $2l-4$ ecuaciones.

Veamos que la matriz de coeficientes es equivalente precisamente a la definida en el Lema 4.6.7 para los casos $p - c_0 = 4l-4$, $4l-6$, $4l-8$ y $x = 3l-3$, $3l-5$, y $3l-7$, respectivamente.

Tenemos (4.16).

Definamos $y_\mu = x_{l-1+\mu}$, con $\mu \geq 1$. Tenemos

$$\begin{aligned} 0 &= \sum_{\mu=1}^{[(j+p-c_0)/2]-(l-1)} (-1)^{l-1+\mu-j} \binom{p-c_0+1-(l-1+\mu)-1}{l-1+\mu-j} y_\mu, \\ &= \sum_{\mu=1}^{[(j+p-c_0)/2]-(l-1)} (-1)^{l-1+\mu-j} \binom{p-c_0-(l-1+\mu)}{l-1+\mu-j} y_\mu \end{aligned}$$

para $j \in [1, l-2] \cup [l+1, 2l-1]$, siendo μ el índice de filas y j el índice de columnas, y teniendo en cuenta que $\binom{a}{b} = \binom{a}{a-b}$, las primeras $l-2$ columnas tienen como coeficiente matricial

$$(-1)^{l-1+\mu-j}\binom{p-c_0+1-l-\mu}{p-c_0+2-2l-2\mu+j}.$$

Ahora, para $j \in [l-1, 2l-3]$ tenemos las siguientes $2l-3-(l-1)+1 = l-1$ columnas correspondientes a $j+2 \in [l+1, 2l-1]$, con coeficiente

$$(-1)^{l-1+\mu-j}\binom{p-c_0+1-l-\mu}{p-c_0+2-2l-2\mu+j+2} = \\ = (-1)^{l-1+\mu-j}\binom{p-c_0+1-l-\mu}{p-c_0+4-2l-2\mu+j}.$$

Consideremos ahora tres casos, correspondientes a

$$p - c_0 = 4l-4, 4l-6, 4l-8.$$

4.6.1. El caso $p - c_0 = 2l - 4$

En lo sucesivo, dados e, u, v números naturales tales que $1 \leq u$, $u+1 < v < e$ denotaremos por $A = A[x, e, u, v] = (a_{ij})_{i,j=1,\ldots,e}$, la matriz definida como sigue:

$$a_{ij} = \begin{cases} \binom{x-i}{t-2i+j} & \text{si } i = 1, \ldots, n_1, \\ 0 & \text{si } i = n_1+1, \ldots, e,\ i-j \neq n_1-u, \\ 1 & \text{si } i = n_1+1, \ldots, e,\ i-j = n_1-u, \end{cases}$$

donde

$$t = e+1-2(v-u-1), \quad n_1 = e-v+u+1.$$

En el caso $p-c_0 = 2l-4$, substituyendo estos valores y eliminando los menos uno de las columnas en donde éstos aparecen, las primeras $l-2$ columnas tienen como coeficiente matricial

$$\binom{p-c_0+1-l-\mu}{p-c_0+2-2l-2\mu+j} = \binom{3l-3-\mu}{2l-2-2\mu+j} = \binom{x-\mu}{t-2\mu+j},$$

con $x = 3l-3$ y $t = 2l-2$. Las siguientes $l-1$ columnas tienen como coeficiente (eliminando los menos uno donde aparezcan)

$$\binom{p-c_0+1-l-\mu}{p-c_0+2-2l-2\mu+j+2} = \binom{3l-3-\mu}{2l-2\mu+j}$$

(observemos que si $j \in [l-1, 2l-3]$ tenemos que $j+2 \in [l+1, 2l-1]$, y podemos aplicar la fórmula).
Para la columna adicional $j = 2l-2$ tenemos

$$\begin{aligned}
0 &= \alpha_{l-1,p-c_0+1} + \alpha_{l,p-c_0+1} \\
&= \alpha_{l-1,4l-3} + \alpha_{l,4l-3} \\
&= \sum_{\mu=1}^{[(l-1+p-c_0)/2]-(l-1)} (-1)^{l-1+\mu-(l-1)} \binom{p-c_0+1-(l-1+\mu)-1}{p-c_0+2-2l-2\mu+(l-1)} y_\mu \\
&\quad + \sum_{\mu=1}^{[(l+p-c_0)/2]-(l-1)} (-1)^{l-1+\mu-l} \binom{p-c_0+1-(l-1+\mu)-1}{p-c_0+2-2l-2\mu+l} y_\mu,
\end{aligned}$$

esto es,

$$\begin{aligned}
&\sum_{\mu=1}^{[(l-1+4l-4)/2]-(l-1)} (-1)^{l-1+\mu-(l-1)} \binom{4l-3-(l-1+\mu)-1}{2l-2-2\mu+(l-1)} y_\mu \\
&\quad + \sum_{\mu=1}^{[(l+4l-4)/2]-(l-1)} (-1)^{l-1+\mu-l} \binom{4l-3-(l-1+\mu)-1}{2l-2-2\mu+l} y_\mu, \\
&= \sum_{\mu=1}^{l-1+[(l-1)/2]} ((-1)^\mu \binom{3l-3-\mu}{2l-2-2\mu+l-1} y_\mu \\
&\quad + (-1)^{\mu-1} \binom{3l-3-\mu}{2l-2-2\mu+l} y_\mu) \\
&= \sum_{\mu=1}^{l-1+[l/2]} (-1)^\mu (\binom{3l-3-\mu}{2l-2-2\mu+l-1} - \binom{3l-3-\mu}{2l-2-2\mu+l}) y_\mu,
\end{aligned}$$

ya que para $l = 2l'+1$ tenemos que $[(l-1)/2] = l' = [l/2]$, y para $l = 2l'$ tenemos que

$$\binom{3l-3-(l-1+l')}{2l-2-2(l-1+l')+l-1} = \binom{3l-3-(l-1+l')}{-1} = 0,$$

de esta manera este coeficiente es

$$(-1)^\mu \left(\binom{3l-3-\mu}{2l-2-2\mu+l-1} - \binom{3l-3-\mu}{2l-2-2\mu+l} \right).$$

Finalmente, eliminando los -1 de las columnas donde aparezcan, tenemos como matriz de coeficientes la matriz D dada en el siguiente Lema para $x = 3l - 3$. Observamos que el determinante de D se reduce al cálculo del determinante de dos matrices con las condiciones del Lema 4.6.7.

Lema 4.6.2. *Sea* $D = (d_{ij})_{1 \leq i,j \leq e-1}$ *una matriz con* $e = 2l - 1$ *y* $t = 2l - 2$ *definidos por*

$$d_{ij} = \binom{x-i}{t-2i+j}, \qquad \textit{para } 1 \leq j \leq l-2,$$

$$d_{ij} = \binom{x-i}{t+2-2i+j}, \qquad \textit{para } l-1 \leq j \leq e-2,$$

$$d_{ij} = \binom{x-i}{t-1-2i+l} - \binom{x-i}{t-2i+l}, \qquad \textit{para } j = e-1,$$

entonces $\det D = \det A[x, 2l-1, l-2, l] + \det A[x, 2l-1, l-1, l+1]$

Demostración. Empecemos con la matriz $Z = (z_{ij})_{1 \leq i,j \leq e}$, definida por

$$z_{ij} = \begin{cases} \binom{x-i}{t-2i+j} & \text{si } i = 1, \ldots, e-1, j = 1, \ldots, e, \\ 0, & \text{si } i = e, j \notin \{l-1, l\}, \\ 1, & \text{si } i = e, j \in \{l-1, l\}. \end{cases}$$

Veamos que $\det Z = \det D$. De cara a relacionar Z con D, hacemos las siguientes transformaciones:

1. A la columna $(l-1)$ de Z, le restamos su columna l. Como consecuencia, obtenemos una matriz Z' con el mismo determinante y los mismos coeficientes que Z, excepto

$$z'_{i,l-1} = z_{i,l-1} - z_{i,l} = \binom{x-i}{t-1-2i+l} - \binom{x-i}{t-2i+l}$$

para $i = 1, \ldots, e-1$, y

$$z'_{e,l-1} = z_{e,l-1} - z_{e,l} = 1 - 1 = 0.$$

2. Trasladamos las dos columnas $l-1$ y l a los dos últimos lugares, esto es, permutamos los dos bloques de columnas ${Z'}^{l-1,l}$ y ${Z'}^{l+1,\ldots,e}$, lo que no origina ningún cambio de signo, y obtenemos la matriz Z'' que tiene la siguiente forma por bloques

$$Z'' = \begin{pmatrix} D & * \\ 0 & 1 \end{pmatrix}.$$

Como consecuencia,

$$\det Z = \det Z' = \det Z'' = \det D.$$

Observemos, finalmente, que si desarrollamos el determinante de Z por la última fila que tiene dos unos en los lugares $(e, l-1), (e, l)$ tenemos

$$\det Z = \det A[x, 2l-1, l-2, l] + \det A[x, 2l-1, l-1, l+1],$$

de donde el resultado deseado. □

4.6.2. El caso $p - c_0 = 4l - 6$

Supongamos que $p - c_0 = 4l - 6$ y $l \geq 3$.
Tenemos

$$0 = \alpha_{j,p-c_0+1} = \alpha_{j,4l-5} = \sum_{\lambda=j}^{[(j+4l-5-1)/2]} (-1)^{\lambda-j} \binom{4l-5-\lambda-1}{\lambda-j} x_\lambda,$$

Consideremos $y_\mu = x_{l-1+\mu}$ para $\mu \geq 1$. Se obtiene que

$$\begin{aligned} 0 &= \sum_{\mu=1}^{[(j+4l-5-1)/2]-(l-1)} (-1)^{l-1+\mu-j} \binom{4l-5-(l-1+\mu)-1}{l-1+\mu-j} y_\mu \\ &= \sum_{\mu=1}^{l-2+[j/2]} (-1)^{l-1+\mu-j} \binom{3l-5-\mu}{l-1+\mu-j} y_\mu \\ &= \sum_{\mu=1}^{l-2+[j/2]} (-1)^{l-1+\mu-j} \binom{3l-5-\mu}{2l-4-2\mu+j} y_\mu \end{aligned}$$

para $j \in [1, l-2] \cup [l+1, 2l-1]$. Tomando μ como índice de filas y j como índice de columnas, y eliminando los signos, deducimos que

$$\binom{3l-5-\mu}{2l-4-2\mu+j} = \binom{x-\mu}{2l-4-2\mu+j},$$

con $x = 3l-5$ and $t = 2l-4$. Ahora, para $j \in [l+1, 2l-1]$ tenemos las siguientes $2l-1-(l+1)+1 = l-1$ que corresponden a $j+2 \in [l-1, 2l-3]$, con coeficiente

$$\binom{3l-5-\mu}{2l-4-2\mu+j+2} = \binom{3l-5-\mu}{2l-2-2\mu+j}.$$

Finalmente, eliminando los -1 de las columnas donde aparezcan, se tiene la matriz D dada en el siguiente Lema para $x = 3l-5$.

Lema 4.6.3. *Consideremos* $D = (d_{ij})_{1 \leq i,j \leq e-2}$ *con* $e = 2l-1$ *y* $t = 2l-4$ *definidas por*

$$d_{ij} = \binom{x-i}{t-2i+j} \qquad \textit{para } 1 \leq j \leq l-2,$$

$$d_{ij} = \binom{x-i}{t+2-2i+j}, \qquad \textit{para } l-1 \leq j \leq e-2,$$

entonces $\det D = \det A[x, 2l-1, l-2, l+1]$

Demostración. Comencemos con la matriz $A = A[x, 2l-1, l-2, l+1]$, podemos escribirla en notación de bloques como

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ 0 & I_2 & 0 \end{pmatrix},$$

donde A_{12} es un bloque de dos columnas situado en los lugares $l-1$ y l, e I_2 es la matriz identidad. Moviendo el bloque formado por las columnas $l-1$, l de A a las dos últimas columnas, obtenemos una matriz on el mismo determinante de A de la forma

$$T = \begin{pmatrix} A_{11} & A_{13} & A_{12} \\ 0 & 0 & I_2 \end{pmatrix}$$

y el determinante de esta matriz es igual al determinante de la siguiente submatriz de tamaño $(e-2) \times (e-2)$:

$$\begin{pmatrix} A_{11} & A_{13} \end{pmatrix}$$

que coincide con la matriz D. □

Nota 4.6.4. Observemos que con el Lema previo se evita el "salto" de dos columnas de la matriz D y unificamos el valor de las casillas (en A) como un único número combinatorio, gracias al orlado de las dos últimas filas

$$(0, I_2, 0).$$

El tamaño de esta matriz aumenta en dos filas como contrapartida, pero unificamos el valor de las casillas y podemos trabajar por operaciones elementales de una manera más elegante.

4.6.3. El caso $p - c_0 = 4l - 8$

Supongamos $p - c_0 = 4l - 8$ y $l \geq 3$.

Tenemos que

$$0 = \alpha_{j,p-c_0+1} = \alpha_{j,4l-7} = \sum_{\lambda=j}^{[(j+4l-7-1)/2]} (-1)^{\lambda-j} \binom{4l-7-\lambda-1}{\lambda-j} x_\lambda.$$

Consideremos $y_\mu = x_{l-1+\mu}$, con $\mu \geq 1$. Tenemos que

$$\begin{aligned} 0 &= \sum_{\mu=1}^{[(j+4l-7-1)/2]-(l-1)} (-1)^{l-1+\mu-j} \binom{4l-7-(l-1+\mu)-1}{l-1+\mu-j} y_\mu \\ &= \sum_{\mu=1}^{l-3+[j/2]} (-1)^{l-1+\mu-j} \binom{3l-7-\mu}{l-1+\mu-j} y_\mu \\ &= \sum_{\mu=1}^{l-3+[j/2]} (-1)^{l-1+\mu-j} \binom{3l-7-\mu}{2l-6-2\mu+j} y_\mu \end{aligned}$$

para $j \in [1, l-2] \cup [l+1, 2l-1]$. Tomando μ como índice de fila y j como índice de columna, eliminando los signos de los columnas obtenemos que las primeras $l-2$ columnas tienen

$$\binom{3l-7-\mu}{2l-6-2\mu+j} = \binom{x-\mu}{t-2\mu+j},$$

con $x = 3l-7$, $t = 2l-6$ como su coeficiente matricial. Ahora, para $j \in [l+1, 2l-1]$ tenemos las siguientes $2l-1-(l+1)+1 = l-1$ columnas

correspondientes a $j+2 \in [l-1, 2l-3]$, con coeficiente

$$\binom{3l-7-\mu}{2l-6-2\mu+j+2} = \binom{3l-7-\mu}{2l-4-2\mu+j}$$

Finalmente, eliminando los -1 de las columnas donde aparecen, obtenemos la matriz D dada en el siguiente Lema para $x = 3l-7$.

Lema 4.6.5. *Consideremos* $D = (d_{ij})_{1 \leq i,j \leq e-2}$ *con* $e = 2l-3$ *y* $t = 2l-4$ *definida por*

$$d_{ij} = \binom{x-i}{t-2i+j}, \qquad \textit{para } 1 \leq j \leq l-2,$$

$$d_{ij} = \binom{x-i}{t+2-2i+j}, \qquad \textit{para } l-1 \leq j \leq e-2,$$

entonces $\det D = \det A[x, 2l-3, l-2, l+1]$.

Demostración. Comencemos con la matriz $A = A[x, 2l-3, l-2, l+1]$, que podemos escribir en notación de bloques como

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ 0 & I_2 & 0 \end{pmatrix},$$

donde A_{12} es un bloque de dos columnas localizado en los lugares $l-1$ y l, e I_2 es la matriz identidad. Moviendo el bloque de las dos columnas $l-1$ y l de A a las dos últimas columnas, se obtiene una matriz con el mismo determinante que A y con la forma

$$T = \begin{pmatrix} A_{11} & A_{13} & A_{12} \\ 0 & 0 & I_2 \end{pmatrix}.$$

El determinante de esta matriz es igual al determinante de la siguiente submatriz de tamaño $(e-2) \times (e-2)$:

$$\begin{pmatrix} A_{11} & A_{13} \end{pmatrix},$$

que coincide con la matriz D. □

Nota 4.6.6. Observemos que, con el Lema previo, evitamos de nuevo el salto de dos columnas (de la matriz D) y unificamos el valor de las casillas (en A) en un único número combinatorio, gracias al orlado de las dos últimas filas con

$$(0, I_2, 0).$$

El tamaño de la matriz es incrementado como contrapartida en dos filas, pero al unificar los valores de las casillas podemos trabajar por operaciones elementales de un modo más elegante.

4.6.4. Los determinantes

De acuerdo con los Lemas anteriores, debemos calcular los determinantes de las siguientes matrices:

$$A[3l-3, 2l-1, l-2, l], \quad x = 3l-3, \quad t = 2l-2;$$

$$A[3l-3, 2l-1, l-1, l+1], \quad x = 3l-3, \quad t = 2l-2;$$

$$A[3l-5, 2l-1, l-2, l+1], \quad A[3l-5, 2l-3, l-2, l+1], \quad x = 3l-5, \quad t = 2l-4;$$

$$A[3l-5, 2l-3, l-2, l+1], \quad A[3l-5, 2l-3, l-2, l+1], \quad x = 3l-5, \quad t = 2l-6;$$

que resolvemos de manera general en el siguiente Lema, teniendo en cuenta que los cuatro casos satisfacen las condiciones

$$t = e + 1 - 2(v - u - 1), \qquad n_1 = e - v + u + 1.$$

En el desarrollo de la demostración de los siguientes resultados, utilizaremos no sólo los índices fundamentales e, u, v, t y n_1, sino también estos otros:

$$i_1 = \left[\frac{t+u}{2}\right], \qquad b_1 = \left[\frac{t-u}{2}\right], \quad b_3 = e - v + 1 - \left[\frac{t-u}{2}\right],$$
$$j_1 = e + u + 1 - \left[\frac{t+u}{2}\right], \quad b_2 = u, \qquad b_4 = v - u - 1.$$

de modo que $j_1 - v = n_1 - i_1$.

Lema 4.6.7. *Sean e, u, v números naturales tales que $1 \leq u$, $u+1 < v < e$. Sea $A = A[x, e, u, v]$.*

Entonces

$$\begin{aligned}\det A =&(-1)^{[u/2]+[(e-v+1)/2]+b_1(e-b_1)+[b_3/2]+(v-1)b_1+[b_1/2]+[b_2/2]+b_3b_4}\\
&\cdot\prod_{s=1}^{u-1}\prod_{j=1}^{s}(2x-t+u-j-s)\cdot\prod_{s=v}^{e-1}\prod_{j=v}^{s}(2x-t+e-j-s)\\
&\cdot\prod_{i=1}^{n_1}\frac{1}{(e+t-2i)!}\cdot\prod_{i=1}^{\left[\frac{t+1}{2}\right]}[x-i,t+1-2i]\\
&\cdot\prod_{i=\left[\frac{t+3}{2}\right]}^{n_1}\frac{1}{[x-(t+1-i),2i-t-1]}\\
&\cdot\prod_{i=i_1+1}^{n_1}(n_1-i)!\cdot\prod_{i=i_1+1}^{n_1}[x-(t+1-i),v-1]\\
&\cdot\prod_{i=1}^{b_1}[n_1-i,b_3]\cdot\prod_{i=1}^{b_1}[b_1+v-1-i,v-1]\cdot\prod_{k=1}^{b_1}(k-1)!\\
&\cdot\prod_{k=1}^{u}(k-1)!\cdot\prod_{i=b_1+1}^{b_1+b_2}[e+t-2i,e-u]\cdot\frac{F(x)}{F(\alpha_0)},\end{aligned}$$

donde

$$\begin{aligned}F(x)=&\prod_{\lambda=\lambda_1}^{\lambda_2-1}(x-\lambda)^{\lambda-\lambda_1+1}\cdot\prod_{\lambda=\lambda_2}^{\lambda_3}(x-\lambda)^{b_1}\cdot\prod_{\lambda=\lambda_3+1}^{\lambda_4-1}(x-\lambda)^{\lambda_4-\lambda+1}\\
&\cdot\prod_{\alpha=\alpha_1}^{\alpha_2-1}(x-\alpha)^{\alpha-\alpha_1+1}\cdot\prod_{\alpha=\alpha_2}^{\alpha_3}(x-\alpha)^{b_4}\cdot\prod_{\alpha=\alpha_3+1}^{\alpha_4}(x-\alpha)^{\alpha_4-\alpha+1},\end{aligned}$$

si $b_4 \leq b_1$ *o, en otro caso,*

$$\begin{aligned}F(x)=&\prod_{\lambda=\lambda_1}^{\lambda_2-1}(x-\lambda)^{\lambda-\lambda_1+1}\cdot\prod_{\lambda=\lambda_2}^{\lambda_3}(x-\lambda)^{b_1}\cdot\prod_{\lambda=\lambda_3+1}^{\lambda_4-1}(x-\lambda)^{\lambda_4-\lambda+1}\\
&\cdot(x-t)\big(x-(t+1)\big),\end{aligned}$$

siendo

$$\lambda_1=3/2,\quad \lambda_2=b_1+1/2,\quad \lambda_3=u+1/2,\quad \lambda_4=u+b_1-1/2$$

and

$$\alpha_1 = t - b_1 + 1, \quad \alpha_2 = t - b_1 + b_4, \quad \alpha_3 = t, \quad \alpha_4 = t + b_4 - 1, \quad \alpha_0 = t - b_1.$$

Demostración. Las transformaciones que vamos a realizar afectarán sólo las primeras n_1 filas y estamos interesados sólo en las alteraciones producidas en los bloques A_{11} y A_{13}, ya que los valores de A_{12} no aparecen en el cálculo del determinante.

Consideremos los índices de columnas p, q, con $1 \leq p < q \leq t+1$, y la submatriz correspondiente $A^{p,\ldots,q}_{1,\ldots,n_1}$. Aplicamos a A las siguientes transformaciones por columnas:

$$A^j \mapsto A^j + 2A^{j+1}, \quad A^j \mapsto \frac{1}{2x - t - j + q - s} A^j$$

para $j = p, \ldots, s$, $s = q-1, q-2, \ldots, p$. Los elementos de la matriz resultante B tienen en sus primeras n_1 filas las mismas componentes que A, excepto

$$b_{ij} = \frac{1}{[x - i + q - j, q - j]} \binom{x + q - j - i}{t + q - 2i}, \quad i = 1, \ldots n_1, \quad j = p, \ldots, q.$$

Si aplicamos sucesivamente este proceso a los bloques formados por las columnas $1, \ldots, u$ y $v, \ldots, e$, obtenemos la matriz $B = (b_{ij})$ cuyas componentes en los bloques correspondientes B_{11} y B_{14} son, respectivamente,

$$b_{ij} = \begin{cases} \frac{1}{[x+u-i-j, u-j]} \binom{x+u-i-j}{t+u-2i}, & \text{si } i = 1, \ldots, t, j = 1, \ldots, u \\ \frac{1}{[x+e-i-j, e-j]} \binom{x+e-i-j}{t+e-2i}, & \text{si } i = 1, \ldots, t, j = v, \ldots, 2l-1. \end{cases}$$

Tenemos la relación entre determinantes

$$\det A = \prod_{s=1}^{u-1} \prod_{j=1}^{s} (2x - t + u - j - s) \cdot \prod_{s=v}^{e-1} \prod_{j=v}^{s} (2x - t + e - j - s) \cdot \det B.$$

Si multiplicamos ahora las filas $i = 1, \ldots, n_1$ por $(e+t-2i)!$, dividimos las filas $i = 1, \ldots, \left[\frac{t+1}{2}\right]$ por $[x-i, t+1-2i]$ y multiplicamos las filas $i = \left[\frac{t+3}{2}\right], \ldots, n_1$ por $[x - (t+1-i), 2i - t - 1]$, obtenemos la matriz $C = (c_{ij})$, con

$$c_{ij} = \begin{cases} [t + e - 2i, e - u][x - (t + 1 - i), j - 1] & \text{para } i = 1, \ldots, n_1, \\ & j = 1, \ldots, u, \\ [x - (t + 1 - i), j - 1] & \text{para } i = 1, \ldots, n_1, \\ & j = v, \ldots, e. \end{cases}$$

Se tiene la siguiente relación entre determinantes:

$$\det B = \prod_{i=1}^{n_1} \frac{1}{(e+t-2i)!} \cdot \prod_{i=1}^{\left[\frac{t+1}{2}\right]} [x-i, t+1-2i] \cdot \prod_{i=\left[\frac{t+3}{2}\right]}^{n_1} \frac{1}{[x-(t+1-i), 2i-t-1]} \cdot \det C.$$

Observemos que $[t+e-2i, e-u] = 0$ a menos que $t+e-2i \geq e-u$, esto es, $i \leq i_1 = \left[\frac{t+u}{2}\right]$. Ahora hacemos algunas transformaciones que triangularizan cada uno de estos dos bloques:

$$C^j \mapsto C^j - (x-t-j+i_1+1)C^{j-1} j = u, u-1, \ldots, s, \quad s = 2, \ldots, u,$$
$$C^j \mapsto C^j - (x-j+n_1-j-1)C^{j-1} j = e, e-1, \ldots, s, \quad s = v+1, \ldots, e,$$

y, cancelando los signos -1 que aparecen explícitamente:

$$C^j \mapsto (-1)^{j-1} C^j j = 1, \ldots, u,$$
$$C^j \mapsto (-1)^{j-v} C^j j = v, \ldots, e.$$

Obtenemos de este modo la matriz teórica $D = (d_{ij})$ cuyas entradas en los bloques correspondientes a A_{11} y A_{13} son

$$d_{ij} = \begin{cases} [i_1 - i, j-1][e+t-2i, e-u], & \text{para } i = 1 \ldots, i_1, \\ & \quad j = 1, \ldots, u, \\ 0 & \text{para } i = i_1 + 1 \ldots, n_1 \\ & \quad j = 1, \ldots, u, \\ [n_1 - i, j-v][x-(t+1-i), v-1], & \text{para } i = 1 \ldots, n_1, \\ & \quad j = v, \ldots, e. \end{cases}$$

Se tiene la siguiente relación entre determinantes:

$$\det C = (-1)^{[u/2]+[(e-v+1)/2]} \cdot \det D.$$

Esta matriz tiene una estructura por bloques de $b_1 + b_2 + b_3 + b_4$ filas $b_2 + b_4 + b_3 + b_1$ columnas:

$$D = \begin{pmatrix} D_{11} & D_{12} & D_{13} & D_{14} \\ 0 & D_{22} & D_{23} & 0 \\ 0 & I & 0 & 0 \end{pmatrix},$$

donde D_{23} es un bloque cuadrado de la forma

$$D_{23} = \begin{pmatrix} * & & & d_{j_1-1,i_1+1} \\ & & \iddots & \\ & d_{v+1,n_1-1} & & \\ d_{v,n_1} & & & 0 \end{pmatrix}.$$

Por tanto,

$$\det D = \prod_{i=i_1+1}^{n_1} (n_1 - i)! \cdot \prod_{i=i_1+1}^{n_1} [x - (t+1-i), v-1] \cdot \det T,$$

con T la matriz teórica siguiente:

$$T = \begin{pmatrix} D_{11} & 0 & 0 & D_{14} \\ 0 & 0 & I_c & 0 \\ 0 & I & 0 & 0 \end{pmatrix}, \quad I_c = \begin{pmatrix} * & & \iddots & 1 \\ & 1 & & \\ 1 & & & 0 \end{pmatrix}.$$

En la fila i-ésima de D_{11} y D_{14} aparecen los factores

$$[e+t-2i, e-u] = 2^{e-u}[e-b_4-i+1/2, k_0] \cdot [e-b_4-i, b_3]$$
$$[n_1-i, j-v] = [e-b_4-i, b_3] \cdot [e-b_4-i-b_3, j-j_1],$$

respectivamente, donde $k_0 = \left[\frac{e-u+1}{2}\right]$.
Dividimos las u columnas de D_{11} entre 2^{e-u} y las filas $i = 1, \ldots, i_1$ entre $[e - b_4 - i, b_3]$. Obtenemos de este modo una matriz $G = (g_{ij})$ cuyas componentes en los bloques correspondientes a D_{11} y D_{14} son

$$g_{ij} = \begin{cases} [i_1 - i, j-1] \cdot [e-b_4-i+1/2, k_0] & \text{para } j = 1, \ldots, u, \\ [e-b_4-i-b_3, j-j_1] \cdot [x-(t+1-i), v-1] & \text{para } j = j_1, \ldots, e. \end{cases}$$

Observamos que las componentes de cada columna de la matriz G de índices $j = 1, \ldots, u$, $j = j_1, \ldots, e$ son polinomios en el índice de fila cuyo grado no es mayor que $(u-1) + k_0$ para $j = 1, \ldots, u$ y $(e - j_1) + (v-1) = u - 1 + k_0$ para $j = j_1, \ldots, t+1$.
Consideremos $\lambda \in \frac{\mathbb{Z}}{2} - \mathbb{Z}$. Tengamos presente los desarrollos

$$[e-b_4-i+1/2, k_0] = \prod_{k=1}^{k_0} (e-b_4-i+3/2-k)$$

$$[\lambda-(t+1-i), v-1] = (-1)^{v-1} \prod_{k=1}^{v-1} (e+1-2b_4-\lambda-i-k).$$

Definimos los conjuntos

$$X = \{e - b_4 - i + 3/2 - k \mid k = 1, \ldots, k_0\},$$

$$Y = \{e + 1 + v - 2b_4 - i - \lambda - k \mid k = 1, \ldots, v - 1\},$$

y sea

$$\{s_1, \ldots s_{q_\lambda}\} = X \cap Y.$$

Tenemos

$$\begin{aligned} &\max X = e - b_4 - i + 1/2, && \min X = e - b_4 - i + 3/2 - k_0, \\ &\max Y = e + v - 2b_4 - i - \lambda, && \min Y = e - 2b_4 + 2 - i - \lambda. \end{aligned}$$

Entonces, para cada índice $i = 1, \ldots, i_1$, las diferencias $s_1 - i, \ldots, s_{q_\lambda} - i$ son factores comunes de la i-ésima fila. De este modo, después de quitar estos factores, las columnas de índices $j = 1, \ldots, u$, $j = j_1, \ldots, e$ en la matriz resultante son polinomios en el índice de fila cuyo grado no supera $u - 1 + k_0 - q_\lambda$. Pero el conjunto de columnas con esta propiedad configura un espacio vectorial de dimensión $u + k_0 - q_\lambda$ y, por tanto, λ es una raíz del determinante de G de multiplicidad mayor o igual que

$$m_\lambda = i_1 - u - k_0 + q_\lambda - 1 = q_\lambda - b_4.$$

El cardinal q_λ de la intersección de X e Y es

$$\min\{\max X - \min Y + 1, \max Y - \min X + 1, k_0, v - 1\},$$

o sea,

$$q_\lambda = \min\{b_4 - 1/2 + \lambda, k_0, u + k_0 + 1/2 - \lambda\},$$

y, de este modo, la multiplicidad de λ en el determinante de la matriz A es mayor o igual que

$$m_\lambda = \min\{\lambda - 1/2, k_0 - b_4, u + k_0 + 1/2 - \lambda - b_4\}.$$

De acuerdo con las condiciones precedentes, definimos:

$$\lambda_1 = 3/2, \quad \lambda_2 = b_1 + 1/2, \quad \lambda_3 = u + 1/2, \quad \lambda_4 = u + b_1 - 1/2.$$

Entonces

$$m_\lambda = \begin{cases} \lambda - \lambda_1 + 1 & \text{si } \lambda_1 \leq \lambda \leq \lambda_2, \\ b_1 & \text{si } \lambda_2 \leq \lambda \leq \lambda_3, \\ \lambda_4 - \lambda + 1 & \text{si } \lambda_3 \leq \lambda \leq \lambda_4. \end{cases}$$

Por otro lado, para cada punto (r, s) tal que $s \geq j_1$ y $r + s \leq i_1 + v$ consideramos el rectángulo

$$R_{(r,s)} = \{(i, j) \mid i \geq r, \quad j \geq s\}.$$

Busquemos los valores $x = \alpha$ que hacen cero todas las componentes g_{ij} de $R_{(r,s)}$. Recordemos que los valores que anulan una columna también anulan las siguientes, por tanto, teniendo en cuenta que las componentes g_{ij} son cero para $i + j > i_1 + j_1$, bastará asegurar que g_{ij} se anula en la parte de la columna s-ésima para $r \leq i \leq i_1 + j_1 - s$, esto es,

$$x - \alpha \in [x - (t + 1 - r), v - 1]' \cap \cdots \cap [x - (t + 1 - (i_1 + j_1 - s)), v - 1]',$$

con lo que

$$[\alpha - (t + 1 - i), v - 1] = 0, \qquad r \leq i \leq i_1 + j_1 - s,$$

y, de este modo,

$$t - 1 - r \leq \alpha \leq t + 1 - r - v + 2, \tag{4.17}$$

$$\ldots, \tag{4.18}$$

$$t + 1 - (i_1 + j_1 - s) \leq \alpha \leq t + 1 - (i_1 + j_1 - s) + v - 2, \tag{4.19}$$

con lo cual

$$t + 1 - r \leq \alpha \leq t + b_r - 1 - (e - s). \tag{4.20}$$

Observemos que si el rectángulo $R_{(r,s)}$ se anula, α es una raíz del determinante de multiplicidad mayor o igual que $(e-r+1)+(e-s+1)-e = e-(r+s)+2$, ya que α anula un rectángulo de ceros con $e-r+1$ filas y $e-s+1$ columnas (véase nota 4.6.8).

Por consiguiente, para cada α calculamos el par (r, s) tal que las desigualdades (4.20) se satisfacen y $r + s$ es mínimo. Por tanto, la multiplicidad de α en el determinante de la matriz T es mayor o igual que m_α, como se detalla en los siguientes casos:

1. $b_4 \leq b_1$:

   a) $t-b_1+1 \leq \alpha \leq t-b_1+b_4$, $r = t+1-\alpha$, $s = j_1$, $m_\alpha = \alpha-(t-b_1)$;
   b) $t-b_1+b_4 \leq \alpha \leq t$, $r = t+1-\alpha$, $s = \alpha-t-b_4+1+e$, $m_\alpha = b_4$;
   c) $t \leq \alpha \leq t+b_4-1$, $r = 1$, $s = \alpha-t-b_4+1+e$, $m_\alpha = t+b_4-\alpha$.

2. $b_4 > b_1$, o, equivalentemente, $b_1 = 1$, $b_4 = 2$:

   a) $\alpha = t$, $m_t = 1$;
   b) $\alpha = t+1$, $m_{t+1} = 1$.

Tenemos, pues:

1. En el caso $b_4 \leq b_1$, si definimos

   $$\alpha_1 = t - b_1 + 1, \quad \alpha_2 = t - b_1 + b_4, \quad \alpha_3 = t, \quad \alpha_4 = t + b_4 - 1,$$

   $\det T$ es divisible por el polinomio

   $$F(x) = \prod_{\lambda=\lambda_1}^{\lambda_2-1} (x-\lambda)^{\lambda-\lambda_1+1} \cdot \prod_{\lambda=\lambda_2}^{\lambda_3} (x-\lambda)^{b_1} \cdot \prod_{\lambda=\lambda_3+1}^{\lambda_4-1} (x-\lambda)^{\lambda_4-\lambda+1} \cdot \prod_{\alpha=\alpha_1}^{\alpha_2-1} (x-\alpha)^{\alpha-\alpha_1+1} \cdot \prod_{\alpha=\alpha_2}^{\alpha_3} (x-\alpha)^{b_4} \cdot \prod_{\alpha=\alpha_3+1}^{\alpha_4} (x-\alpha)^{\alpha_4-\alpha+1},$$

   cuyo grado es

   $$(\lambda_2-\lambda_1+1)(\lambda_3-\lambda_1+1)+(\alpha_2-\alpha_1+1)(\alpha_3-\alpha_1+1) = b_1u+b_4b_1 = b_1(v-1).$$

2. El caso $b_4 > b_1$ sólo se da cuando $b_1 = 1$ y $b_4 = 2$ y las raíces $\alpha_1 = t$ y $\alpha_2 = t+1$ son simples. Por tanto, el determinante de la matriz T es divisible por el polinomio

   $$F(x) = \prod_{\lambda=\lambda_1}^{\lambda_2-1} (x-\lambda)^{\lambda-\lambda_1+1} \cdot \prod_{\lambda=\lambda_2}^{\lambda_3} (x-\lambda)^{b_1} \cdot \prod_{\lambda=\lambda_3+1}^{\lambda_4-1} (x-\lambda)^{\lambda_4-\lambda+1} \cdot (x-t) \cdot \big(x-(t+1)\big),$$

   cuyo grado es

   $$(\lambda_2-\lambda_1+1)(\lambda_3-\lambda_1+1)+2 = b_1u+b_4b_1 = b_1(v-1).$$

Por otro lado, el grado del determinante de T es a lo sumo $(e-j_1+1)(v-1) = b_1(v-1)$. Como consecuencia, resulta que el determinante de T difiere de este polinomio en un factor constante, $\det T = KF(x)$.

De cara a calcular el valor de la constante K, calculamos el valor del determinante en el punto $\alpha_0 = t - b_1$, de modo que

$$\det T = \frac{\det T(\alpha_0)}{F(\alpha_0)} \cdot F(x).$$

Este valor satisface las condiciones (4.20) y anula el rectángulo R_{b_1+1,j_1}:

$$t + 1 - (b_1 + 1) \leq \alpha_0 \leq t + b_4 - 1 - (e - j_1)$$

Por tanto, la matriz $T(\alpha_0)$ puede expresarse en bloques de $b_1 + b_2 + b_3 + b_4$ filas y $b_2 + b_4 + b_3 + b_1$ columnas en la forma

$$T = \begin{pmatrix} T_{11} & 0 & 0 & T_{14} \\ T_{21} & 0 & 0 & 0 \\ 0 & 0 & I_c & 0 \\ 0 & I & 0 & 0 \end{pmatrix},$$

donde

$$T_{21} = \begin{pmatrix} * & & & d_{i_1-u+1,u} \\ & & \iddots & \\ & d_{i_1-1,2} & & \\ d_{i_1,1} & & & 0 \end{pmatrix}$$

y

$$T_{14} = (d_{ij}(\alpha_0))_{i=1,\dots b_1, \quad j=j_1,\dots,e}.$$

Como T_{21} es una matriz regular, podemos permutar los bloques de columnas, y obtenemos

$$\begin{aligned} \det T(\alpha_0) &= (-1)^{b_1(e-b_1)+b_3b_4} \det \begin{pmatrix} T_{14} & T_{11} & 0 & 0 \\ 0 & T_{21} & & \\ & & I_c & \\ & & & I \end{pmatrix} \\ &= (-1)^{b_1(e-b_1)+b_3b_4+[b_3/2]} \det T_{14} \ \det T_{21}. \end{aligned}$$

Más aún,

$$\det T_{21} = (-1)^{[b_2/2]} \prod_{k=1}^{u} (k-1)! \cdot \prod_{i=b_1+1}^{b_1+b_2} [e + t - 2i, e - u].$$

En T_{14}, extraemos de cada fila el factor $[n_1 - i, j_1 - v][\alpha_0 - (t+1-i), v-1] = [n_1 - i, b_3](-1)^{v-1}[b_1 + v - 1 - i, v - 1]$. Por tanto,

$$\det T_{14}(\alpha_0) = (-1)^{(v-1)b_1} \prod_{i=1}^{b_1} [n_1 - i, b_3] \cdot \prod_{i=1}^{b_1} [b_1 + v - 1 - i, v - 1] \cdot \det S_{14},$$

donde

$$S_{14} = (s_{ij}(\alpha_0))_{i=1,\ldots b_1, \quad j=j_1,\ldots,e}, \qquad s_{ij} = [n_1 - i - j_1 + v, j - j_1].$$

Trasladando los índices de columnas, obtenemos

$$S_{14} = (\alpha_{ij})_{i,j=1,\ldots,b_1}, \quad \alpha_{ij} = [a - i, j - 1], \quad a = n_1 - j_1 + v.$$

Por inducción, se ve fácilmente que el determinante de esta última matriz es

$$\det S_{14} = (-1)^{[b_1/2]} \prod_{k=1}^{b_1} (k-1)!. \quad \square$$

Nota 4.6.8. Supongamos que α es una raíz del determinante de una matriz $M(x)$ de orden w cuyos términos son polinomios en x. Entonces:

1. Si $\operatorname{rank} M(\alpha) = w - m$ entonces α es una raíz de $\det M(x)$ de multiplicidad no menor que m.

2. Si $M(\alpha)$ tiene un bloque rectangular de ceros de a filas y b columnas, entonces α es una raíz de $\det M(x)$ de multiplicidad mayor o igual que $a + b - 2$.

Teorema 4.6.9. *Supongamos que $p - c_0 = 4l - 4 - 2\omega$, $\omega = 0,\ 1,\ 2$ y $l \geq 3$. Entonces*

$$2c \geq m - p - 2l + c_0 - 1.$$

Demostración. Si $c_0 \leq 2l - 1$, entonces, por [22], sabemos que $2c \geq m - p - 2l + c_0 - 1$. Supongamos, por reducción al absurdo, que se da lo contrario, es decir, $p + 2l - c_0 + 1 \leq m - 2c - 1$. Entonces $c_0 \geq 2l$, y, por tanto, $p = c_0 + 4l - 4 - 2\omega \geq 6l - 4 - 2\omega$. De acuerdo con los Lemas previos, obtenemos un sistema de $2l - 2 - 2\omega + 1$ ecuaciones con $2l - 2 - 2\omega + 1$

incógnitas con $y_1 = x_l \neq 0$, de modo que el determinante de la matriz de coeficientes es no nulo y p divide a uno de los factores de

$$\det A[x, 2l-1, l-2, l] + \det A[x, 2l-1, l-1, l+1], \tag{4.21}$$

con $x = 3l-3$, en el caso $\omega = 0$, o a uno de los factores de

$$\det A[x, 2l-1, l-2, l+1], \tag{4.22}$$

con $x = 3l-5$, en el caso $\omega = 1$, o a uno de los factores de

$$\det A[x, 2l-3, l-2, l+1], \tag{4.23}$$

en el caso $x = 3l-7$; ahora, cada factor que aparece en (4.21), (4.22) o (4.23) pertenece al intervalo $(0, 6l-9)$ y p es mayor que $6l-4$ en el caso (4.21); pertenece al intervalo $\big(0, 2(3l-5)-3\big) = (0, 6l-13)$ y p es mayor que $6l-6$ en el caso (4.22), y pertenece al intervalo $\big(0, 2(3l-7-3)\big) = (0, 6l-17)$ y p es mayor que $6l-8$ en el caso (4.23), en cualquier caso, una contradicción. □

Teorema 4.6.10. *Supongamos que* $c_0 \in \{p-4l+3\lambda+1, p-4l+3\lambda+3, p-4l+3\lambda+5\}$ *con* $\lambda \geq 1$. *Entonces*

$$2c \geq m - p - 2l + c_0 - 1.$$

Demostración. Por inducción sobre $\lambda + |G|$. Para $\lambda = 1$ tenemos que

$$c_0 \in \{p-4l+4, p-4l+6, p-4l+8\}$$

y el resultado ha sido probado. Supongamos que $\lambda \geq 2$ y que se tiene la relación

$$c_0 \in \{p-4l+3\lambda+1, p-4l+3\lambda+3, p-4l+3\lambda+5\}.$$

Trabajando con el p-grupo de clase maximal H, tenemos que $c_0(H) = c_0+1$, $l(H) = l-1$ y

$$c_0(H) = c_0 + 1 \in \{p-4l+3\lambda+2, p-4l+3\lambda+4, p-4l+3\lambda+6\}$$

esto es,

$$c_0(H) \in \{\{p-4(l-1)+3(\lambda-1)+1, p-4(l-1)+3\lambda+3, p-4(l-1)+3(\lambda-1)+5\}$$

luego por la hipótesis de inducción aplicada a H concluimos que $2(c+1) = 2c(H) \geq m-1-p-2(l-1)+(c_0+1)-1$, de donde la desigualdad esperada. □

Las tablas 4.20 a 4.25 esquematizan los resultados probados en relación al conjunto de valores posibles de c_0 y l.

$p = 17$	1	2	3	4	5	6	7
0		6	8	10	12	14	16
1			9	11	13	15	16
2				12	14	15	29
3					14	26	29
4				13	23	26	27
5			12	20	23	25	26
6					22	23	25
7				19	21	23	24
8				18	20	21	23
9			15	17	19	20	22
10				16	18	19	21
11			13	15	18	18	20
12			12	18	15	17	19
13		9	18	12	14	16	18
14		18	9	11	13	15	17
15	18	6	8	10	12	14	16

Cuadro 4.20: Cotas probadas para $p = 17$

$p = 19$	1	2	3	4	5	6	7	8
0		6	8	10	12	14	16	18
1			9	11	13	15	17	18
2				12	14	16	17	33
3					15	16	30	33
4					15	27	30	31
5				14	24	27	29	30
6				21	24	26	27	29
7				21	23	25	27	28
8					22	24	25	27
9				19	21	23	24	26
10				18	20	21	23	25
11			15	17	19	21	22	24
12				16	18	20	21	23
13			13	15	20	18	20	22
14			12	20	15	17	19	21
15		9	20	12	14	16	18	20
16		20	9	11	13	15	17	19
17	20	6	8	10	12	14	16	18

Cuadro 4.21: Cotas probadas para $p = 19$

$p = 23$	1	2	3	4	5	6	7	8	9	10
0		6	8	10	12	14	16	18	20	22
1			9	11	13	15	17	19	21	22
2				12	14	16	18	20	21	41
3					15	17	19	20	38	41
4						18	19	35	38	39
5						18	32	35	37	38
6					17	29	32	34	35	37
7				16	26	29	31	33	35	36
8						28	30	32	33	35
9					25	27	29	31	32	34
10					24	26	28	29	31	33
11				21	23	25	27	29	30	32
12					22	24	26	27	29	31
13				19	21	23	25	26	28	30
14				18	20	22	24	25	27	29
15			15	17	19	21	24	24	26	28
16				16	18	24	21	23	25	27
17			13	15	24	18	20	22	24	26
18			12	24	15	17	19	21	23	25
19		9	24	12	14	16	18	20	22	24
20		24	9	11	13	15	17	19	21	23
21	24	6	8	10	12	14	16	18	20	22

Cuadro 4.22: Cotas probadas para $p = 23$

$p = 29$	1	2	3	4	5	6	7	8	9	10	11	12	13
0		6	8	10	12	14	16	18	20	22	24	26	28
1			9	11	13	15	17	19	21	23	25	27	28
2				12	14	16	18	20	22	24	26	27	53
3					15	17	19	21	23	25	26	50	53
4						18	20	22	24	25	47	50	51
5							21	23	24	44	47	49	50
6								23	41	44	46	47	49
7							22	38	41	43	45	47	48
8						21	35	38	40	42	44	45	47
9					20	32	35	37	39	41	43	44	46
10							34	36	38	40	41	43	45
11						31	33	35	37	39	41	42	44
12						30	32	34	36	38	39	41	43
13					27	29	31	33	35	37	38	40	42
14						28	30	32	34	35	37	39	41
15					25	27	29	31	33	35	36	38	40
16					24	26	28	30	32	33	35	37	39
17				21	23	25	27	29	31	32	34	36	38
18					22	24	26	28	30	31	33	35	37
19				19	21	23	25	27	30	30	32	34	36
20				18	20	22	24	30	27	29	31	33	35
21			15	17	19	21	30	24	26	28	30	32	34
22				16	18	30	21	23	25	27	29	31	33
23			13	15	30	18	20	22	24	26	28	30	32
24			12	30	15	17	19	21	23	25	27	29	31
25		9	30	12	14	16	18	20	22	24	26	28	30
26		30	9	11	13	15	17	19	21	23	25	27	29
27	30	6	8	10	12	14	16	18	20	22	24	26	28

Cuadro 4.23: Cotas probadas para $p = 29$

$p = 31$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0		6	8	10	12	14	16	18	20	22	24	26	28	30
1			9	11	13	15	17	19	21	23	25	27	29	30
2				12	14	16	18	20	22	24	26	28	29	57
3					15	17	19	21	23	25	27	28	54	57
4						18	20	22	24	26	27	51	54	55
5							21	23	25	26	48	51	53	54
6								24	25	45	48	50	51	53
7								24	42	45	47	49	51	52
8							23	39	42	44	46	48	49	51
9						22	36	39	41	43	45	47	48	50
10						33	36	38	40	42	44	45	47	49
11						33	35	37	39	41	43	45	46	48
12							34	36	38	40	42	43	45	47
13						31	33	35	37	39	41	42	44	46
14						30	32	34	36	38	39	41	43	45
15					27	29	31	33	35	37	39	40	42	44
16						28	30	32	34	36	37	39	41	43
17					25	27	29	31	33	35	36	38	40	42
18					24	26	28	30	32	33	35	37	39	41
19				21	23	25	27	29	31	33	34	36	38	40
20					22	24	26	28	30	32	33	35	37	39
21				19	21	23	25	27	32	30	32	34	36	38
22				18	20	22	24	32	27	29	31	33	35	37
23			15	17	19	21	32	24	26	28	30	32	34	36
24				16	18	32	21	23	25	27	29	31	33	35
25			13	15	32	18	20	22	24	26	28	30	32	34
26			12	32	15	17	19	21	23	25	27	29	31	33
27		9	32	12	14	16	18	20	22	24	26	28	30	32
28		32	9	11	13	15	17	19	21	23	25	27	29	31
29	32	6	8	10	12	14	16	18	20	22	24	26	28	30

Cuadro 4.24: Cotas probadas para $p = 31$

$p = 41$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0		6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
1			9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	40
2				12	14	16	18	20	22	24	26	28	30	32	34	36	38	39	77
3					15	17	19	21	23	25	27	29	31	33	35	37	38	74	77
4						18	20	22	24	26	28	30	32	34	36	37	71	74	75
5							21	23	25	27	29	31	33	35	36	68	71	73	74
6								24	26	28	30	32	34	35	65	68	70	71	73
7									27	29	31	33	34	62	65	67	69	71	72
8										30	32	33	59	62	64	66	68	69	71
9											32	56	59	61	63	65	67	68	70
10										31	53	56	58	60	62	64	65	67	69
11									30	50	53	55	57	59	61	63	65	66	68
12								29	47	50	52	54	56	58	60	62	63	65	67
13							28	44	47	49	51	53	55	57	59	61	62	64	66
14									46	48	50	52	54	56	58	59	61	63	65
15								43	45	47	49	51	53	55	57	59	60	62	64
16								42	44	46	48	50	52	54	56	57	59	61	63
17							39	41	43	45	47	49	51	53	55	56	58	60	62
18								40	42	44	46	48	50	52	53	55	57	59	61
19							37	39	41	43	45	47	49	51	53	54	56	58	60
20							36	38	40	42	44	46	48	50	51	53	55	57	59
21						33	35	37	39	41	43	45	47	49	50	52	54	56	58
22							34	36	38	40	42	44	46	47	49	51	53	55	57
23						31	33	35	37	39	41	43	45	47	48	50	52	54	56
24						30	32	34	36	38	40	42	44	45	47	49	51	53	55
25					27	29	31	33	35	37	39	41	43	44	46	48	50	52	54
26						28	30	32	34	36	38	40	42	43	45	47	49	51	53
27					25	27	29	31	33	35	37	39	42	42	44	46	48	50	52
28					24	26	28	30	32	34	36	42	39	41	43	45	47	49	51
29				21	23	25	27	29	31	33	42	36	38	40	42	44	46	48	50
30					22	24	26	28	30	42	33	35	37	39	41	43	45	47	49
31				19	21	23	25	27	42	30	32	34	36	38	40	42	44	46	48
32				18	20	22	24	42	27	29	31	33	35	37	39	41	43	45	47
33			15	17	19	21	42	24	26	28	30	32	34	36	38	40	42	44	46
34				16	18	42	21	23	25	27	29	31	33	35	37	39	41	43	45
35			13	15	42	18	20	22	24	26	28	30	32	34	36	38	40	42	44
36			12	42	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43
37		9	42	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42
38		42	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41
39	42	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40

Cuadro 4.25: Cotas probadas para $p = 41$

$l = 1$	x_2/x_1	x_3/x_1	x_4/x_1	x_5/x_1	x_6/x_1	x_7/x_1
$c_0 = 0$	0	0	0	0	0	0
$c_0 = 9$	15	5	3	8	4	12
$c_0 = 10$	6	7	13	15	11	2
	0	0	10	11	16	9
$c_0 = 11$	0	0	13	1	10	3
$c_0 = 12$	0	0	1	0	6	3
	6	6	6	11	4	3
	14	7	7	*	$13x_1 + 2x_5$	$10x_1 + 15x_5$
$c_0 = 13$	9	9	8	0	0	2
$c_0 = 14$	1	16	0	0	0	0
$c_0 = 15$	0	0	0	0	0	0
	16	0	0	0	0	0

Cuadro 4.26: Soluciones para $p = 17$, $l = 1$

4.7. Un ejemplo de las soluciones para $p = 17$

En las tablas 4.26 a 4.31 mostramos soluciones no nulas al sistema de ecuaciones de Jacobi para el nivel anterior al dado en las tablas. La entrada $*$ significa que todos los valores posibles de esta variable están permitidos. Si la entrada es una forma en x_i, esto quierer decir que la variable en el numerador de la primera fila es igual a esa forma, y si aparece únicamente un número, esto significa que la fracción en la primera fila es igual a este número (módulo p). Mostramos todas las soluciones halladas por nuestro algoritmo. Notemos que, para algunas de las filas, en particular para $l = 1$, $1 \leq c_0 \leq 13$ ó para $l = 2$, $c_0 = 4$, nuestro algoritmo nos da sólo una cota superior, pero de momento no podemos asegurar que ésta sea la mejor cota (una cota exacta, en nuestra terminología).

$l=2$	x_3/x_2	x_4/x_2	x_5/x_2	x_6/x_2	x_7/x_2
$c_0=0$	$*$	$*$	$*$	$*$	$*$
$c_0=1$	3	2	$*$	$*$	$*$
$c_0=2$	7	1	10	$*$	$*$
$c_0=3$	9	15	3	2	$*$
$c_0=6$	3	15	15	15	10
$c_0=7$	2	15	4	7	11
$c_0=8$	2	15	4	7	11
$c_0=9$	3	6	5	4	12
	10	14	5	5	16
$c_0=10$	0	2	6	15	14
$c_0=11$	6	3	$*$	$*$	$11x_2+11x_5+2x_6$
$c_0=12$	15	15	$*$	$*$	$*$
$c_0=13$	16	$*$	$*$	$*$	$*$
$c_0=14$	0	0	0	0	0
$c_0=15$	$*$	$*$	$*$	$*$	$*$

Cuadro 4.27: Soluciones para $p=17$, $l=2$

$l = 3$	x_4/x_3	x_5/x_3	x_6/x_3	x_7/x_3
$c_0 = 0$	$*$	$*$	$*$	$*$
$c_0 = 1$	6	$*$	$*$	$*$
$c_0 = 2$	11	10	$*$	$*$
$c_0 = 3$	12	3	10	$*$
$c_0 = 4$	0	6	16	15
	10	12	11	16
$c_0 = 5$	16	5	2	5
	7	7	7	13
	0	1	9	4
$c_0 = 6$	2	15	4	7
$c_0 = 7$	2	15	4	7
$c_0 = 8$	10	14	5	5
$c_0 = 9$	$*$	$2x_3 + 16x_4$	13	$12x_3 + 9x_4$
$c_0 = 10$	6	3	3	$*$
$c_0 = 11$	15	1	$*$	$*$
$c_0 = 12$	16	$*$	$*$	$*$
$c_0 = 13$	0	0	0	0
$c_0 \geq 14$	$*$	$*$	$*$	$*$

Cuadro 4.28: Soluciones para $p = 17$, $l = 3$

$l = 4$	x_5/x_4	x_6/x_4	x_7/x_4
$c_0 = 0$	$*$	$*$	$*$
$c_0 = 1$	10	$*$	$*$
$c_0 = 2$	4	8	$*$
$c_0 = 3$	13	7	1
$c_0 = 4$	12	4	8
$c_0 = 5$	2	15	4
$c_0 = 6$	2	15	4
$c_0 = 7$	10	14	5
$c_0 = 8$	$*$	$2x_4 + 16x_5$	13
$c_0 = 9$	$*$	3	16
$c_0 = 10$	15	1	$*$
$c_0 = 11$	16	0	$*$
$c_0 = 12$	0	0	0
$c_0 \geq 13$	$*$	$*$	$*$

Cuadro 4.29: Soluciones para $p = 17$, $l = 4$

$l = 5$	x_6/x_5	x_7/x_5
$c_0 = 0$	$*$	$*$
$c_0 = 1$	15	$*$
$c_0 = 2$	3	10
$c_0 = 3$	4	14
$c_0 = 4$	2	15
$c_0 = 5$	2	15
$c_0 = 6$	10	14
$c_0 = 7$	12	7
$c_0 = 8$	13	3
$c_0 = 9$	15	1
$c_0 = 10$	16	0
$c_0 = 11$	0	0
$c_0 \geq 12$	$*$	$*$

Cuadro 4.30: Soluciones para $p = 17$, $l = 5$

$l = 6$	x_7/x_6
$c_0 = 0$	$*$
$c_0 = 1$	4
$c_0 = 2$	8
$c_0 = 3$	2
$c_0 = 4$	2
$c_0 = 5$	10
$c_0 = 6$	$*$
$c_0 = 7$	13
$c_0 \geq 8$	$*$

Cuadro 4.31: Soluciones para $p = 17$, $l = 6$

Apéndice A

Listados de los programas utilizados

A.1. Algoritmo de cálculo de configuraciones de ceros y no ceros y Jacobi sin consideración de números primos

A.1.1. El archivo `bullet.h`

```
/* Archivo bullet.h, con cabeceras generales del programa bullet.c */
#define TRUE 1
#define FALSE 0
#define MAX_STACK_OVERFLOW 511


#define MAX_NUM_DE_CASILLAS 72
/* esto es 8*9, */
#define COEFMAX 32
/* con 16 habria bastante para las x, para las z necesitaremos casi el */
/* doble.  Esto duplica el uso de memoria.*/
#define MAXFILAS 32



/* Cabeceras de funciones sobre primos que aparecen en la */
/* descomposicion de numeros */
/* leidas en primos.c*/

int log_2_int(long int);

int es_primo(long int n);
```

```
long int mayor_primo(long int n);

/* Funciones para ser leidas por fraccion.c.  Cabeceras */
/* relacionadas con operaciones con fracciones. */


typedef struct {
  long int num;
  long int den;
} fraccion;

fraccion sum(fraccion, fraccion);
fraccion menos(fraccion);
fraccion simplifica(fraccion);
fraccion prod(fraccion, fraccion);


fraccion inversa(fraccion);

long int  gcd(long int,long int);

fraccion leefraccion(FILE *archivo);

void escribe_fraccion(fraccion);

/* Cabeceras leidas por casilla.c, con las definiciones */
/* relativas a las casillas del triangulo T_G */

typedef int indice_de_casilla[2];
typedef indice_de_casilla lista_de_casillas[MAX_NUM_DE_CASILLAS];

typedef struct
  {
    unsigned char longitud;
    fraccion coef[COEFMAX];
  } casilla;

casilla prodesc(fraccion, casilla);

casilla sum_cas(casilla, casilla);

casilla menos_cas(casilla);

int es_cero(casilla);

void escribe_casilla(casilla);

int son_casillas_iguales(casilla cas1, casilla cas2);

long int mayor_primo_casilla(casilla);

/* Cabeceras asociadas a triang.c, que  define las */
/* funciones asociadas a los triangulos T_G. */

typedef casilla triang[MAXFILAS][COEFMAX];

void  copia_triangulo(triang *,triang *);
```

```
void escribe_bullets(triang,int, FILE *);

void escribe_bullets_uno(triang, int, FILE *);

void escribe_triangulo(triang,int);

void aumenta_fila(triang *tr, int fila);

/* Funciones de pila de bulletero, para ser leidas por pila.c */

struct PILA{
  triang *triangulo;
  int num_no_ceros;
  lista_de_casillas lista;
  int posicion;
  struct PILA *padre;
} ;

void anyadir_a_pila(triang);

/* Prototipos de funciones relacionadas con */
/* la substitucion de unas casillas en otras dentro de los triangulos */
/* del tipo T_G.  Para ser leido por despeje.c */


typedef struct
{
  casilla cas;
  int pos;
} despeje;

casilla substituir(casilla cs, despeje desp);

despeje despejar_una_variable(casilla c);

/* Funciones de salida de bulletero */

void escribe_salida(triang,int, FILE *);

void anyadir_a_lista_no_ceros(lista_de_casillas milista, int numero,
      int f, int c);

void verifica(int valor);
void lee_triangulo(triang *, int, FILE *);
void lee_casilla(casilla *, int, FILE *);
void escribe_triangulo_archivo(triang, int, FILE *);
void escribe_casilla_archivo(casilla, int, FILE*);
void escribe_casilla_archivo_TeX(casilla cas, char *nom_var, FILE
 *archivo);
void escribe_casilla_archivo_maple(casilla cas, FILE *archivo);
void escribe_primera_fila_TeX(triang, int, FILE*);
void escribe_lista_archivo_maple(triang tr, int fila, FILE *archivo);

/* Generacion de arboles de bulletero */

void copia_lista(lista_de_casillas,lista_de_casillas);
```

```
int calcula_la_lista(struct PILA *pila, int);
void arbol(struct PILA *pila, int fila, FILE *, FILE *, FILE *, FILE *);

typedef casilla *zeta_vector;

int numero_de_ceros(triang tr, int fila);
int zetas_iguales(zeta_vector *, int);
int iniciar_zetas(zeta_vector *z, indice_de_casilla **z_cas, triang
  tr, int fila, int c_cero);

unsigned long int choose(unsigned long int x1, unsigned long int x2);

void aplicar_lema_dos_uno(int mat[], int long_mat, int a, int b);

void iniciar_lema_dos_uno(int mat[], int long_mat);

void lema_dos_uno(zeta_vector *z, indice_de_casilla **z_cas,
  zeta_vector *zeta,  triang  tr, int fila, int
  c_cero, int num_de_zetas, FILE *salida);

void substituye_rapido_casilla(casilla *cas, int mat[]);

int despejar_una_variable_en_lista(zeta_vector *z, indice_de_casilla
   **z_cas, zeta_vector *zeta,
   triang tr, int fila, int c_cero,
   int num_de_zetas, FILE *salida,
   FILE *salida_log, FILE
   *salida_log2);

void substituir_una_variable_en_lista(zeta_vector *z,
      indice_de_casilla **z_cas,
      zeta_vector *zeta, triang tr,
      int fila, int c_cero, int
      num_de_zetas, int num_de_var,
      despeje des, FILE *salida);

#define vermem(xx) if ((xx)==NULL) {\
      fprintf(stderr,"Error de memoria\n");\
exit(1);\
}
```

A.1.2. El archivo `bulletvl.h`

```
/* Archivo bulletvl.h, con cabeceras generales del programa bulletvl.c */

#define TRUE 1
#define FALSE 0
/*#define MAX_STACK_OVERFLOW 1024*/


#define MAX_NUM_DE_CASILLAS 72
/* esto es 8*9, */
#define COEFMAX 32
/* con 16 habria bastante para las x, para las z necesitaremos casi el */
/* doble.  Esto duplica el uso de memoria.*/
```

```
#define MAXFILAS 32
#define MAXPR     (longint2vl(65537))

/* Cabeceras de funciones con un nuevo tipo "verylong", entero de */
/* longitud tan grande como se desee.  Este valor viene determinado */
/* por la macro MAYOR.*/

#define MAYOR 8

typedef struct {
  int sgn;
  unsigned long int coef[MAYOR];
} verylong;

void escribevl(verylong);
void escribevl_archivo(FILE *, verylong);
verylong longint2vl(unsigned long int);
verylong sumavl(verylong a, verylong b);
verylong restavl(verylong a, verylong b);
int es_cerovl(verylong a);
int es_mayorvl(verylong a, verylong b);
int es_igualvl(verylong a, verylong b);
verylong menosvl(verylong a);
verylong absvl(verylong a);
void representa(verylong a);
verylong prodvl(verylong a, verylong b);
verylong div_intvl(verylong a, unsigned long int b);
unsigned long int mod_intvl(verylong a, unsigned long int b);
verylong divvl(verylong a, verylong b);
verylong modvl(verylong a, verylong b);
verylong gcdvl(verylong a, verylong b);


/* Cabeceras de funciones sobre primovls que aparecen en la */
/* descomposicion de numeros */
/* leidas en primovls.c*/

int log_2_int(long int);

int es_primovl(verylong n);

verylong mayor_primovl(verylong n, verylong maxpr);

/* Funciones para ser leidas por fraccionvl.c.  Cabeceras */
/* relacionadas con operaciones con fraccionvles. */


typedef struct {
  verylong num;
  verylong den;
} fraccionvl;


fraccionvl sumfrvl(fraccionvl, fraccionvl);
fraccionvl menosfrvl(fraccionvl);
fraccionvl simplificafrvl(fraccionvl);
fraccionvl prodfrvl(fraccionvl, fraccionvl);
```

```
fraccionvl inversafrvl(fraccionvl);

verylong gcdvl(verylong, verylong);

fraccionvl leefraccionvl(FILE *archivo);

void escribe_fraccionvl(fraccionvl);

/* Cabeceras leidas por casillavl.c, con las definiciones */
/* relativas a las casillavls del triangulo T_G */


typedef struct
  {
    unsigned char longitud;
    fraccionvl coef[COEFMAX];
  } casillavl;


casillavl prodescvl(fraccionvl escalar, casillavl cas);

casillavl sum_casvl(casillavl, casillavl);

casillavl menos_casvl(casillavl);

int es_cero_casillavl(casillavl);


void escribe_casillavl(casillavl);

int son_casillavls_iguales(casillavl cas1, casillavl cas2);

verylong mayor_primo_casillavl(casillavl);

/* Cabeceras asociadas a triang.c, que  define las */
/* funciones asociadas a los triangulos T_G. */




/* Prototipos de funciones relacionadas con */
/* la substitucion de unas casillavls en otras dentro de los triangulos */
/* del tipo T_G.  Para ser leido por despeje.c */


typedef struct
{
  casillavl cas;
  int pos;
} despejevl;

casillavl substituirvl(casillavl cs, despejevl desp);

despejevl despejar_una_variablevl(casillavl c);
```

```
/* Funciones de salida de bulletero */

void lee_casillavl(casillavl *, int, FILE *);
void escribe_casillavl_archivo(casillavl, int, FILE*);
void escribe_casillavl_archivo_TeX(casillavl cas, char *nom_var, FILE
 *archivo);
void escribe_casillavl_archivo_maple(casillavl cas, FILE *archivo);

/* Generacion de arboles de bulletero */


typedef casillavl *zeta_vectorvl;

int zetas_igualesvl(zeta_vectorvl *, int);

int iniciar_zetasvl(zeta_vectorvl *z, indice_de_casilla **z_cas, triang
  tr, int fila, int c_cero);

void lema_dos_unovl(zeta_vectorvl *z, indice_de_casilla **z_cas,
  zeta_vectorvl *zeta,  triang  tr, int fila, int
  c_cero, int num_de_zetas, FILE *salida);
void substituye_rapido_casillavl(casillavl *cas, int mat[]);

int despejar_una_variablevl_en_lista(zeta_vectorvl *z, indice_de_casilla
   **z_cas, zeta_vectorvl *zeta,
   triang tr, int fila, int c_cero,
   int num_de_zetas, FILE *salida,
   FILE *salida_log, FILE
   *salida_log2);


void substituir_una_variablevl_en_lista(zeta_vectorvl *z,
      indice_de_casilla **z_cas,
      zeta_vectorvl *zeta, triang tr,
      int fila, int c_cero, int
      num_de_zetas, int num_de_var,
      despejevl des, FILE *salida);

#define vermem(xx) if ((xx)==NULL) {\
      fprintf(stderr,"Error de memoria\n");\
exit(1);\
}

int mainvl(triang *tr, int tamanyo, indice_de_casilla **z_cas, FILE
   *salida, FILE *salida_log, FILE *salida_log2);
```

A.1.3. El archivo `Makefile`

```
zcc=cc -Wall
OBJETOS=arbol.o despeje.o fraccion.o casilla.o  primos.o  triang.o\
 output.o
OBJETOSVL=$(OBJETOS) despejevl.o fraccionvl.o casillavl.o  primosvl.o \
gros.o outputvl.o zetavl.o zeta2.o
```

```
cabeceras= bullet.h
all:    bullet zeta jacobi

arbol.o: arbol.c  $(cabeceras)
$(cc) -c  -g arbol.c

despeje.o: despeje.c  $(cabeceras)
$(cc) -c  -g despeje.c

fraccion.o: fraccion.c  $(cabeceras)
$(cc) -c  -g fraccion.c

casilla.o: casilla.c   $(cabeceras)
$(cc) -c  -g casilla.c

output.o:  output.c $(cabeceras)
$(cc) -c  -g output.c

primos.o:  primos.c $(cabeceras)
$(cc) -c  -g primos.c

triang.o:  triang.c $(cabeceras)
$(cc) -c  -g triang.c

bullet.o: bullet.c $(cabeceras)
$(cc) -c  -g bullet.c

bullet: bullet.o $(OBJETOS) $(cabeceras)
$(cc) -o bullet  -g bullet.o $(OBJETOSVL)

zeta: zeta.o zeta2.o $(OBJETOSVL) $(cabeceras)
$(cc) -o zeta -g zeta.o $(OBJETOSVL)

zeta.o: zeta.c $(cabeceras)
$(cc) -c -g zeta.c

zeta2.o: zeta2.c $(cabeceras)
$(cc) -c -g zeta2.c

gros.o: gros.c bulletvl.h
$(cc) -c -g gros.c

fraccionvl.o: fraccionvl.c bulletvl.h
$(cc) -c -g fraccionvl.c


casillavl.o: casillavl.c bulletvl.h
$(cc) -c -g casillavl.c

zetavl.o: zetavl.c bulletvl.h
$(cc) -c -g zetavl.c

outputvl.o: outputvl.c bulletvl.h
$(cc) -c -g outputvl.c

primosvl.o: primosvl.c bulletvl.h
$(cc) -c -g primosvl.c
```

```
despejevl.o: despejevl.c bulletvl.h
$(cc) -c -g despejevl.c

jacobi: jacobi1.o $(OBJETOS) $(OBJETOSVL)

jacobi1.o: jacobi1.c bulletvl.h bullet.h
$(cc) -c -g jacobi1.c
```

A.1.4. El archivo `arbol.c`

```
/* arbol.c */
/* Desarrolla un arbol de bulletero construido con anterioridad. */
#include <stdio.h>
#include <stdlib.h>
#include "bullet.h"

extern long int CONTADOR, LONG_FILA_TeX;

void copia_lista(lista_de_casillas original,lista_de_casillas copia)
{
  int i,j;
  for (i=0;i<MAX_NUM_DE_CASILLAS;i++)
    {
      for (j=0;j<2;j++)
{
  copia[i][j]=original[i][j];
  };
    };
}

int calcula_la_lista(struct PILA *pila, int fila)
{
  int i,j,res;

  res=0;
  for (i=0; i<fila-1; i++)
    {
      for (j=0; j<=i/2; j++)
{
  if (!es_cero((*(pila->triangulo))[i][j]))
    {
      anyadir_a_lista_no_ceros(pila->lista, res, i, j);
      res++;
    };
};
    }
  return res;
}


void arbol(struct PILA *pila, int fila, FILE *archivo_salida, FILE
   *archivo_salida_TeX, FILE *archivo_cab_TeX, FILE
   *archivo_maple)
{
```

```
  int forzado_cero, fila_maxima, columna,
  ultima_columna_valida, ultima_columna_substituible, i, j;
  triang nuevo_triangulo;
  despeje para_substituir;
  struct PILA *nueva_pila;
/*  long int p;*/

  if (fila%2)
    {
      ultima_columna_valida=(fila-1)/2;
      ultima_columna_substituible=(fila-1)/2;
    }
  else
    {
      ultima_columna_valida=(fila-4)/2;
      ultima_columna_substituible=(fila-2)/2;
    };
  fila_maxima=fila-1;
  for(columna=(*pila).posicion; columna<=ultima_columna_valida;
      columna++)
    {
      copia_triangulo(pila->triangulo,&nuevo_triangulo);
      if (!es_cero(nuevo_triangulo[fila_maxima][columna]))
{
  /* Despejamos una variable, segun la funcion */
  /* despejar_una_variable */

  para_substituir=despejar_una_variable(nuevo_triangulo
[fila_maxima][columna]);

  /* Se substituye la variable despejada con anterioridad en */
  /* las posiciones no nulas del triangulo.  Este proceso se */
  /* hace hasta acabar con todas las posiciones no nulas del */
  /* triangulo o forzar un cero, controlado por la variable */
  /* forzado_cero.*/
  forzado_cero=0;

  for(i=0;
      i<pila->num_no_ceros
/*       &&!es_cero(nuevo_triangulo[pila->lista[i][0]]
 [pila->lista[i][1]]) */
      && !forzado_cero;
      i++)
    {
      nuevo_triangulo[pila->lista[i][0]]
[pila->lista[i][1]]=
  substituir(nuevo_triangulo[pila->lista[i][0]]
     [pila->lista[i][1]],
     para_substituir);
      if (es_cero(nuevo_triangulo[pila->lista[i][0]]
  [pila->lista[i][1]]))
{
  forzado_cero=1;
};
    };
  for (j=0;j<=ultima_columna_substituible;j++)
    {
```

```
      nuevo_triangulo[fila_maxima][j]=
substituir(nuevo_triangulo[fila_maxima][j],
   para_substituir);
    };
  if (!forzado_cero)
    {
      CONTADOR++;
        escribe_salida(nuevo_triangulo,fila,archivo_salida);
      escribe_bullets(nuevo_triangulo,fila,
      archivo_salida_TeX);
      escribe_primera_fila_TeX(nuevo_triangulo, fila,
       archivo_cab_TeX);
      escribe_lista_archivo_maple(nuevo_triangulo, fila,
  archivo_maple);
      vermem(nueva_pila=(struct PILA*) malloc(sizeof(struct PILA)));
      /* vermem(nueva_pila->padre=(struct PILA*) */
      /* malloc(sizeof(struct PILA)));*/
      vermem(nueva_pila->triangulo=(triang *) malloc(sizeof(triang)));
      copia_triangulo(&nuevo_triangulo, (nueva_pila->triangulo));
      copia_lista(pila->lista, nueva_pila->lista);
      nueva_pila->posicion=columna+1;
      nueva_pila->num_no_ceros=pila->num_no_ceros;
/*       nueva_pila->primo=p;*/
      (*nueva_pila).padre=pila;
      arbol(nueva_pila,fila, archivo_salida,
    archivo_salida_TeX, archivo_cab_TeX,
    archivo_maple); /* Llamada recursiva */
      free(nueva_pila->triangulo);
      free(nueva_pila);
      anyadir_a_lista_no_ceros(pila->lista, pila->num_no_ceros,
       fila_maxima, columna);
      pila->num_no_ceros++;
    };
};/* fin de if (!es_cero ...) */
    };
};
```

A.1.5. El archivo `despeje.c`

```
/* Archivo despeje.c, incluye funciones relacionadas con la */
/* substitucion de unas casillas en otras dentro de los triangulos */
/* del tipo T_G. */

#include <stdio.h>
#include <stdlib.h>
#include "bullet.h"

extern int OVERFLOW;

casilla substituir(casilla cs, despeje desp)
```

```
{
  casilla temp,res,simp, resp;
  int i, ind;
  long int mcd, mcm, mcm1;
  fraccion frtemp, fr;
  if (cs.longitud!=desp.cas.longitud)
    {
      printf("casilla y despeje de diferente longitud");
      exit(1);
    };
  if (es_cero(cs))
    {
      res=cs;
    }
  else
    {
      /*  printf("CALCULANDO: cs.coef[desp.pos]=(%d/%d), "
  "desp.cas.coef[desp.pos]=(%d/%d), frtemp=(%d/%d)\n",
  cs.coef[desp.pos].num, cs.coef[desp.pos].den,
  desp.cas.coef[desp.pos].num, desp.cas.coef[desp.pos].den,
  frtemp.num, frtemp.den);
  */
      ind=0;
      while (cs.coef[ind].num==0)
{
  ind++;
};
      mcd=cs.coef[ind].num;
      mcm=cs.coef[ind].den;
      for (i=ind+1;i<cs.longitud;i++)
{
  if (cs.coef[i].num!=0)
    {
      mcd=gcd(cs.coef[i].num,mcd);
      mcm1=(cs.coef[i].den/gcd(mcm,cs.coef[i].den));
      if (log_2_int(mcm1)+log_2_int(mcm) > 30)
{
  OVERFLOW=1;
  return(cs);
}
      else
{
  mcm=mcm1*mcm;
};
    };
};
      fr.num=mcm;
      fr.den=mcd;
      simp=prodesc(fr,cs);
      if (OVERFLOW)
{
  return(cs);
};

      frtemp=prod(simp.coef[desp.pos],
  menos(inversa(desp.cas.coef[desp.pos])));
      if (OVERFLOW)
```

```
{
  return(cs);
};
      temp=prodesc(frtemp,desp.cas);
      if (OVERFLOW)
{
  return(cs);
};
      resp=sum_cas(temp,simp);
      if (OVERFLOW)
{
  return(cs);
};
      res=prodesc(inversa(fr),resp);
      if (OVERFLOW)
{
  return(cs);
};
    }

  return(res);
}

/* El criterio que seguimos para despejar una variable es considerar */
/* el numero de variable que tiene _menor_ su mayor_primo, tanto del */
/* numerador, como del denominador.  En caso de empate con este */
/* criterio, se substituira la variable mas a la  derecha. */

despeje despejar_una_variable(casilla c)
{
  despeje res/*, res2*/;
  int i,j;
  long int mp, mpc;

  res.pos=0;
  res.cas=c;
  j=0;
  while (!c.coef[j].num)
    {
      j++;
    };
  res.pos=j;
  mp=mayor_primo(c.coef[j].num);
  if (OVERFLOW)
    {
      return(res);
    };
  res.cas=c;
  for (i=j+1;i<c.longitud; i++)
    {
      if (c.coef[i].num!=0)
{
  mpc=mayor_primo(c.coef[i].num);
  if (OVERFLOW)
    {
      return(res);
    };
```

```
  if (mpc<=mp)
    {
      mp=mpc;
      res.pos=i;
    };
};
    };
  return (res);
}
```

A.1.6. El archivo `fraccion.c`

```
#include <stdio.h>
#include <stdlib.h>
#include "bullet.h"

extern long int CONTADOR;
extern int OVERFLOW, STACK_OVERFLOW;

int log_2_int(long int n)
{
  int j;
  long int k, num;

  if ((n) && !(2*n))
    {
      fprintf(stderr,"n=-2147483648 es un valor no aceptable.\n");
      return(31);
    }
  else
    {
      if (n==0)
{
  return(0);
}
      else
{
  k=((n<0) ? -n : n);
  num=1;
  j=0;
  while((num<k) && (j<=30))
    {
      num*=2;
      j++;
    };
  return(j);
};
    };
}
```

```
/* La siguiente funcion suma dos fracciones. */
fraccion sum(fraccion primero, fraccion segundo)
{
  fraccion res, res2;
  int an, ad, bn, bd;

  an=log_2_int(primero.num);
  ad=log_2_int(primero.den);
  bn=log_2_int(segundo.num);
  bd=log_2_int(segundo.den);

  if ((ad+bd>30) || (an+bd>=30) || (bn+ad>=30))
    {
      fprintf(stderr, "Posible desbordamiento en suma de"
      " fracciones (CONTADOR=%ld).\n", CONTADOR);
      OVERFLOW=1;
    };

  res.num=primero.num*segundo.den+primero.den*segundo.num;
  res.den=primero.den*segundo.den;
  res2=simplifica(res);
  return(res2);
}

/* Calculo del opuesto de una fraccion */

fraccion menos(fraccion fr)
{
  fraccion res;
  res.num=-(fr.num);
  res.den=fr.den;
  return(simplifica(res));
}


/* Funcion disenyada para simplificar una fraccion */
fraccion simplifica(fraccion fr)
{
  long int mcd;
  fraccion res;
  if (fr.num==0)
    {
      res.num=0;
      res.den=1;
    }
  else
    {
      mcd=gcd(fr.num,fr.den);
      res.num=fr.num/mcd;
      res.den=fr.den/mcd;
    };
  if (res.den<0)
    {
      res.num=-res.num;
      res.den=-res.den;
    };
```

```
  return res;
}

fraccion prod(fraccion primero, fraccion segundo)
{
  fraccion t1, t2, t3, t4, t5, t6, res, res2;

  t1=simplifica(primero);
  t2=simplifica(segundo);
  t3.num=t1.num;
  t3.den=t2.den;
  t4.num=t2.num;
  t4.den=t1.den;
  t5=simplifica(t3);
  t6=simplifica(t4);
  if ((log_2_int(t5.num) + log_2_int(t6.num) > 30) &&
      (log_2_int(t5.den) + log_2_int(t6.den) > 30))
    {
      fprintf(stderr, "Posible desbordamiento en producto de"
      " fracciones (CONTADOR=%ld).\n", CONTADOR);
      OVERFLOW=1;
    };

  res.num=t5.num*t6.num;
  res.den=t5.den*t6.den;
  res2=simplifica(res);
  return (res2);
}

fraccion inversa(fraccion fr)
{
  fraccion res;
  if (fr.num==0)
    {
      fprintf ( stderr, "Error en funcion \"inversa\": No puedo"
       "dividir por cero");
      exit(1);
    }
  else
    {
      res.num=fr.den;
      res.den=fr.num;
      return(res);
    };
}

long int gcd(long int a,long int b)
{
  /* Nuestro objetivo aqui es aplicar el algoritmo de Euclides para el */
  /* calculo del mcd de dos numeros enteros.*/

  long int d;

  STACK_OVERFLOW++;
  if (STACK_OVERFLOW>MAX_STACK_OVERFLOW)
    {
      fprintf(stderr, "Pila desbordada.\n");
```

```
      exit(1);
    };

  if (( (a!=0) && (!(2*a)) ) || ( (b!=0) && (!(2*b)) ))
    {
      fprintf(stderr, "Posible desbordamiento entero en funci\363n gcd.\n");
      OVERFLOW=1;
      return(1);
    };
  if (a<0)
    {
      d=gcd(-a,b);
      STACK_OVERFLOW--;
      return(d);
    }
  else
    {
      if (b<0)
{
  d=gcd(a,-b);
  STACK_OVERFLOW--;
  return(d);
}
      else
{
  if (b==0)
    {
      STACK_OVERFLOW--;
      return (a);
    }
  else
    {
      d=gcd(b,a%b);
      STACK_OVERFLOW--;
      return(d);
    };
};
    };
}

fraccion leefraccion(FILE *archivo)
{
  long int n,d;
  fraccion res;
  fscanf(archivo, "(%ld/%ld)",&n,&d);
  res.num=n;
  res.den=d;
  return(res);
}


void escribe_fraccion(fraccion fr)
{
  printf("(%ld/%ld)", fr.num,fr.den);
}
```

A.1.7. El archivo `casilla.c`

```
/* Archivo casilla.c */
/* Operaciones con casillas del triangulo T_G */

#include <stdio.h>
#include <stdlib.h>
#include "bullet.h"

extern int OVERFLOW;

casilla prodesc(fraccion escalar, casilla cas)
{
  unsigned char i;
  casilla res;
  res.longitud=cas.longitud;
  for (i=0; i<cas.longitud; i++)
    {
      res.coef[i]=prod(escalar, cas.coef[i]);
      if (OVERFLOW)
{
  return(cas);
};
    };
  return(res);
}

casilla menos_cas(casilla cas)
{
  fraccion menosuno;
  menosuno.num=-1;
  menosuno.den=1;
  return(prodesc(menosuno,cas));
}

casilla sum_cas(casilla cas1, casilla cas2)
{
  unsigned char i;
  casilla res;
  if (cas1.longitud!=cas2.longitud)
    {
      fprintf(stderr, "Error, solo esta definida la suma de"
      "\"casillas\" de igual longitud");
      exit(1);
    }
  else
    {
      res.longitud=cas1.longitud;
      for (i=0;i<cas1.longitud;i++)
{
  res.coef[i]=sum(cas1.coef[i],cas2.coef[i]);
  if (OVERFLOW)
    {
      return(cas1);
    };
}
      return (res);
```

```
    };
}


int es_cero(casilla cas)
{
  int escero;
  int i;
  escero=1;
  for (i=0;
       i<cas.longitud;
       i++)
    {
      if (escero)
{
  escero=(cas.coef[i].num==0);
};
    };
  return(escero);
}



void escribe_casilla(casilla c)
{
  int i;
  for (i=0;i<c.longitud;i++)
    {
      escribe_fraccion(c.coef[i]);
    };
}

int son_casillas_iguales(casilla cas1, casilla cas2)
{
  int i, res;
  if (cas1.longitud!=cas2.longitud)
    {
      return 0;
    }
  else
    {
      res=1;
      for (i=0; (i<cas1.longitud) && res; i++)
{
  res= ((cas1.coef[i].num==cas2.coef[i].num) &&
(cas1.coef[i].den==cas2.coef[i].den));
};
      return(res);
    };
}
```

A.1.8. El archivo `output.c`

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
#include "bullet.h"

extern long int CONTADOR;

void verifica(int valor)
{
  if (valor<0)
    {
      perror("Error de escritura");
      exit(1);
    };
}

void lee_triangulo(triang *tr, int fila, FILE *archivo)
{
  int i,j, longitud;

  longitud=(fila+1)/2;
  for (i=fila-1;i>=0;i--)
    {
      for (j=0;j<=(i/2);j++)
{
  lee_casilla(&((*tr)[i][j]), longitud, archivo);
};
    };
}

void lee_casilla(casilla *cas, int longitud, FILE *archivo)
{
  int j;

  cas->longitud=longitud;
  for (j=0; j<longitud; j++)
    {
      if (!fscanf(archivo,"(%ld/%ld)", &(cas->coef[j].num), &(cas->coef[j].den)))
{
  printf ("Error %c\n", '\007');
};
    };
  fscanf(archivo," ");
}
void escribe_casilla_archivo(casilla cas, int longitud, FILE *archivo)
{
  int j;

  cas.longitud=longitud;
  for (j=0; j<longitud; j++)
    {
      verifica(fprintf(archivo,"(%ld/%ld)", cas.coef[j].num,
       cas.coef[j].den));
    };
  verifica(fprintf(archivo," "));
}
```

```
void escribe_triangulo_archivo(triang tr, int fila, FILE *archivo)
{
  int i,j, longitud;

  longitud=(fila+1)/2;
  for (i=fila-1;i>=0;i--)
    {
      for (j=0;j<=(i/2);j++)
{
  escribe_casilla_archivo(tr[i][j], longitud, archivo);
};
      verifica(fprintf(archivo,"\n"));
    };
  verifica(fprintf(archivo,"\n"));
}

void escribe_casilla_archivo_TeX(casilla cas, char *nom_var, FILE
 *archivo)
{
  int j;
  int salio_primer_elemento;

  if (es_cero(cas))
    {
      verifica(fprintf(archivo, "0"));
    }
  else
    {
      salio_primer_elemento=0;
      for (j=0; j<cas.longitud; j++)
{
  if (cas.coef[j].num)
    {
      if (cas.coef[j].den!=1)
{
  if (cas.coef[j].num>0)
    {
      if (!salio_primer_elemento)
{
  verifica(fprintf(archivo,"(%ld/%ld)%s_{%d}",
   cas.coef[j].num,
   cas.coef[j].den, nom_var,
   j+1));
  salio_primer_elemento=1;
}
      else /* ya salio primer elemento */
{
  verifica(fprintf(archivo,
   "+(%ld/%ld)%s_{%d}",
   cas.coef[j].num,
   cas.coef[j].den, nom_var,
   j+1));
};
    }
  else /* cas.coef[j]<0 */
    {
      verifica(fprintf(archivo,"-(%ld/%ld)%s_{%d}",
```

```
       -cas.coef[j].num,
       cas.coef[j].den, nom_var,
       j+1));
      salio_primer_elemento=1;
    }
}
      else /* cas.coef[j].den=1, esto es, es entero */
{
  if (cas.coef[j].num>0)
    {
      if (!salio_primer_elemento)
{
  if (cas.coef[j].num!=1)
    {
      verifica(fprintf(archivo, "%ld%s_{%d}",
       cas.coef[j].num,
       nom_var, j+1));
      salio_primer_elemento=1;
    }
  else /* cas.coef[j].num=1 */
    {
      verifica(fprintf(archivo, "%s_{%d}",
       nom_var, j+1));
      salio_primer_elemento=1;
    };
}
      else /* salio_primer_elemento */
{
  if (cas.coef[j].num!=1)
    {
      verifica(fprintf(archivo, "+%ld%s_{%d}",
       cas.coef[j].num,
       nom_var, j+1));
    }
  else /* cas.coef[j].num==1 */
    {
      verifica(fprintf(archivo, "+%s_{%d}",
       nom_var, j+1));
      salio_primer_elemento=1;
    }
}
    }
  else /* cas.coef[j].num<0 */
    {
      if (cas.coef[j].num!=-1)
{
  verifica(fprintf(archivo, "%ld%s_{%d}",
   cas.coef[j].num, nom_var,
   j+1));
}
      else /* cas.coef[j].num==-1 */
{
  verifica(fprintf(archivo, "-%s_{%d}",
   nom_var, j+1));
  salio_primer_elemento=1;
};
    };
```

```
  salio_primer_elemento=1;
};
    };
};
    };
  verifica(fprintf(archivo, " "));
}

void escribe_primera_fila_TeX(triang tr, int fila, FILE *archivo)
{
  int i, j, longitud;
  long int p,q;

  longitud=(fila+1)/2;
  verifica(fprintf(archivo, "\\primerafila{%ld:", CONTADOR));
  p=1;
  for (i=fila-1;i==fila-1;i--) /* con i>=0 se escribe todo el */
       /* triangulo*/
    {
      for (j=0;j<=(i/2);j++)
{
  verifica(fprintf(archivo, " & "));
  escribe_casilla_archivo_TeX(tr[i][j], "x", archivo);
  q=mayor_primo_casilla(tr[i][j]);
  if (q>p)
    {
      p=q;
    };
};
    };
  verifica(fprintf(archivo, " & p>%ld \\cr}\n",p));
}

void escribe_lista_archivo_maple(triang tr, int fila, FILE *archivo)
{
  int i,j, k;
  lista_de_casillas lista_de_ceros;
  int num_de_ceros;

  num_de_ceros=0;
  verifica(fprintf(archivo, "# %ld\n", CONTADOR));
  for (i=0;i<=fila-1;i++)
    {
      for (j=0; j<=(i/2); j++)
{
  if (es_cero(tr[i][j]))
    {
      lista_de_ceros[num_de_ceros][0]=i;
      lista_de_ceros[num_de_ceros][1]=j;
      num_de_ceros++;
    }
};
    };
  verifica(fprintf(archivo, "liston:=["));
  for (k=0;k<num_de_ceros;k++)
    {
      if (k>0)
```

```
{
  verifica(fprintf(archivo, ", "));
};
      /* Notemos que no son iguales los datos exigidos para la salida */
      /* Maple que para la salida de aqui. */
      verifica(fprintf(archivo, "[%d,%d]", lista_de_ceros[k][1]+1,
       lista_de_ceros[k][0]+2-lista_de_ceros[k][1]));
    };

  verifica(fprintf(archivo,"];\n\n"));
}

void escribe_salida(triang tr, int fila, FILE *archivo)
{
  escribe_triangulo_archivo(tr, fila, archivo);
}

void anyadir_a_lista_no_ceros(lista_de_casillas milista, int numero,
      int f, int c)
{
  milista[numero][0]=f;
  milista[numero][1]=c;
}
```

A.1.9. El archivo `primos.c`

```
/* primos.c */
/* Funciones sobre primos que aparecen en la descomposicion de numeros */

#include <stdio.h>
#include "bullet.h"

extern int OVERFLOW, STACK_OVERFLOW;

int es_primo(long int n)
{
  return (n==mayor_primo(n));
}

long int mayor_primo(long int n)
{
  long int m,p, mp, temp;

  STACK_OVERFLOW++;
  if (STACK_OVERFLOW > MAX_STACK_OVERFLOW)
    {
      fprintf(stderr, "Desbordamiento de pila en mayor_primo.\n");
      exit(1);
    };
  if ((n!=0) && (!(2*n)))
    {
      fprintf(stderr, "Desbordamiento en funci\363n mayor_primo.\n");
      OVERFLOW=1;
      STACK_OVERFLOW--;
```

```
      return(65537);
    };

  if (n<0)
    {
      temp=mayor_primo(-n);
      STACK_OVERFLOW--;
      return(temp);
    }
  else
    {
      if (!n)
{
  STACK_OVERFLOW--;
  return 0L;
}
      else
{
  mp=1L;
  m=n;
  p=2L;
  while (!(m%p))
    {
      m/=p;
      mp=p;
    };
  p=1L;
  while ((m>1L) && (p*p<=m))
    {
      p+=2L;
      while (!(m%p))
{
  m/=p;
  mp=p;
};
    };
  if ((p*p>m) && (m>1L))
    {
      mp=m;
    };
  STACK_OVERFLOW--;
  return mp;
};
    };
}

long int mayor_primo_casilla(casilla cas)
{
  int i;
  long int m,p;

  if (es_cero(cas))
    {
      return((long) 1);
    }
  else
    {
```

```
      p=1;
      for (i=0;i<cas.longitud;i++)
{
  m=mayor_primo(cas.coef[i].num);
  if (OVERFLOW)
    {
      return(p);
    };
  if (m>p)
    {
      p=m;
    };
};
      return(p);
    }
}
```

A.1.10. El archivo `triang.c`

```
/* triang.c */
/* Define las funciones asociadas a los triangulos T_G. */

/* La siguiente funcion copia el triangulo original en el copia. */
/* Supondremos que ha habido suficiente espacio para que "quepa" todo */
/* el triangulo. */

#include <stdio.h>
#include <stdlib.h>
#include "bullet.h"

extern long int CONTADOR, LONG_FILA_TeX;

void  copia_triangulo(triang *original, triang *copia)
{
  int i,j;
  for (i=0;i<MAXFILAS;i++)
    {
      for (j=0;j<=(i/2);j++)
{
  (*copia)[i][j]=(*original)[i][j];
};
    };
}

void escribe_bullets(triang tr, int fila, FILE *salida_TeX)
{
  if (CONTADOR==1)
    {
      verifica(fprintf(salida_TeX, "\\fila{\n"));
    };
  escribe_bullets_uno(tr, fila, salida_TeX);
  if (CONTADOR%LONG_FILA_TeX)
    {
      verifica(fprintf(salida_TeX, "&\n"));
    }
```

```
  else
    {
      verifica(fprintf(salida_TeX,"\\cr}\n\\fila{"));
    };
}

void escribe_bullets_uno(triang tr, int fila, FILE *salida_TeX)
{
  int i,j;

  verifica(fprintf(salida_TeX, "\\taula{\n"));

  for (i=fila-1;i>=0;i--)
    {
      for (j=0;j<=(i/2);j++)
{
  if (es_cero(tr[i][j]))
    {
      verifica(fprintf(salida_TeX,"0"));
    }
  else
    {
      verifica(fprintf(salida_TeX,"\\bullet"));
    };
  if (j<=(i/2)-1)
    {
      verifica(fprintf(salida_TeX, "&"));
    };
};
      verifica(fprintf(salida_TeX,"\\cr\n"));
    };
  verifica(fprintf(salida_TeX,"}{%ld}",CONTADOR));
}

void aumenta_fila(triang *tr, int fila)
{
  int i,j;
  casilla cas;
  if (fila%2) /* fila impar */
    {
      /* introducimos una nueva variable */
      for (i=0;i<fila-1;i++)
{
  for (j=0;j<=(i/2);j++)
    {
      (*tr)[i][j].longitud=(fila+1)/2;
      (*tr)[i][j].coef[(fila-1)/2].num=0;
      (*tr)[i][j].coef[(fila-1)/2].den=1;
    };
};
      /* definimos el x_k correspondiente */
      for (j=0;j<(fila-1)/2;j++)
{
  cas.longitud=(fila+1)/2;
  cas.coef[j].num=0;
  cas.coef[j].den=1;
}
```

```
      cas.coef[(fila-1)/2].num=1;
      cas.coef[(fila-1)/2].den=1;
      (*tr)[fila-1][(fila-1)/2]=cas;
      /* Aplicamos la propiedad de Bernoulli */
      for (j=(fila-3)/2;j>=0;j--)
{
  (*tr)[fila-1][j]=sum_cas((*tr)[fila-2][j],menos_cas((*tr)[fila-1][j+1]));
};
    }
  else
    {
      /* copiamos una casilla */
      (*tr)[fila-1][(fila-2)/2]=(*tr)[fila-2][(fila-2)/2];
      /* aplicamos la propiedad de Bernoulli */
      for (j=(fila-4)/2;j>=0;j--)
{
  (*tr)[fila-1][j]=sum_cas((*tr)[fila-2][j],menos_cas((*tr)[fila-1][j+1]));
};
    };
}

void escribe_triangulo(triang tr, int fila)
{
  int i,j;
  for (i=fila-1;i>=0;i--)
    {
      for (j=0;j<=(i/2);j++)
{
  if (j>0)
    {
      printf(" & \t");
    };
  if (es_cero(tr[i][j]))
    {
      printf("0");
    }
  else
    {
      escribe_casilla(tr[i][j]);
    };
};
      printf("\\cr\n");
    };
  printf("\n");
}
```

A.1.11. El archivo `bullet.c`

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <fcntl.h>
#include <string.h>
#include <sys/stat.h>
```

```
#include "bullet.h"

long int CONTADOR;
long int LONG_FILA_TeX;
int STACK_OVERFLOW, OVERFLOW;

int main(int argc, char *argv[])
{
  struct PILA *pila;
  FILE *entrada, *salida, *salida_TeX, *salida_cab_TeX, *salida_maple;
  int tamanyo;
  char *nombre_salida;
  char *nombre_salida_TeX;
  char *nombre_cab_TeX;
  char *nombre_salida_maple;

  if (argc<3)
    {
      fprintf(stderr, "N\372mero de argumentos muy peque\361o.\n");
      exit(1);
    };
  vermem(pila=(struct PILA *) malloc(sizeof(struct PILA)));
  vermem((*pila).triangulo=(triang*) malloc(sizeof(triang)));

  if ((entrada=fopen(argv[1], "rt"))==NULL)
    {
      perror("Error en archivo de entrada");
    }
  else
    {
      if (fscanf(entrada, "%d ", &tamanyo)<=0)
{
  perror("Error de lectura en archivo de entrada");
  exit(1);
};
      tamanyo++;
      nombre_salida=argv[2];
      vermem(nombre_salida_TeX=(char *)
malloc(sizeof(char)*(strlen(nombre_salida)+4)));
      strcpy(nombre_salida_TeX, nombre_salida);
      strcat(nombre_salida_TeX, ".tex");

      vermem(nombre_cab_TeX=(char *)
malloc(sizeof(char)*(strlen(nombre_salida)+7)));
      strcpy(nombre_cab_TeX, nombre_salida);
      strcat(nombre_cab_TeX, "cab.tex");

      vermem(nombre_salida_maple=(char *)
malloc(sizeof(char)*(strlen(nombre_salida)+6)));
      strcpy(nombre_salida_maple, nombre_salida);
      strcat(nombre_salida_maple, ".maple");

      CONTADOR=0;
      LONG_FILA_TeX=14;

      if ((salida=fopen(nombre_salida, "wt"))==NULL)
{
```

```
  perror("Error en archivo de salida");
  exit(1);
};
      if ((salida_TeX=fopen(nombre_salida_TeX, "wt"))==NULL)
{
  perror("Error en archivo de salida TeX");
  exit(1);
};
      if ((salida_cab_TeX=fopen(nombre_cab_TeX, "wt"))==NULL)
{
  perror("Error en archivo de salida de cabeceras TeX");
  exit(1);
};
      if ((salida_maple=fopen(nombre_salida_maple, "wt"))==NULL)
{
  perror("Error en archivo de salida de listas Maple");
  exit(1);
};
      verifica(fprintf(salida, "%d\n\n", tamanyo));
      while (!feof(entrada))
{
  lee_triangulo(pila->triangulo,tamanyo-1, entrada);


/*   (*pila).primo=1;*/
  aumenta_fila(pila->triangulo,tamanyo);
  CONTADOR++;
  escribe_salida(*(pila->triangulo),tamanyo,salida);
  escribe_bullets(*(pila->triangulo),tamanyo, salida_TeX);
  escribe_primera_fila_TeX(*(pila->triangulo), tamanyo,
   salida_cab_TeX);
  escribe_lista_archivo_maple(*(pila->triangulo), tamanyo,
      salida_maple);
  pila->num_no_ceros=calcula_la_lista(pila,tamanyo);
  arbol(pila, tamanyo, salida, salida_TeX, salida_cab_TeX,
salida_maple);
}
    }
  fclose(entrada);
  fclose(salida);
  verifica(fprintf(salida_TeX, "\\cr}"));
  fclose(salida_TeX);
  fclose(salida_cab_TeX);
  fclose(salida_maple);
  free(pila->triangulo);
  free(pila);
  free(nombre_salida_TeX);
  free(nombre_cab_TeX);
  free(nombre_salida_maple);
  return(0);
}
```

A.1.12. El archivo `zeta.c`

```
/* Archivo zeta.c */
/* Contiene funciones para argumentar con los z_k en los teoremas */
/* sobre los $p$-grupos de clase maximal. */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "bullet.h"
#include "bulletvl.h"

long int CONTADOR;
long int LONG_FILA_TeX;
int OVERFLOW, STACK_OVERFLOW;

int main(int argc, char **argv)
{
  int se_puede_despejar;
  zeta_vector *z, *zeta;
  indice_de_casilla **z_cas;
  triang *tr;
  int tamanyo, num_de_alphas_nulas, i, num_de_ceros;
  FILE *entrada, *salida, *salida_log, *salida_log2;
  char *nombre_salida, *nombre_salida_log, *nombre_salida_log2;

  OVERFLOW=0;
  if ((entrada=fopen(argv[1], "rt"))==NULL)
    {
      perror("Error en archivo de entrada");
    }
  else
    {
      if (fscanf(entrada, "%d ", &tamanyo)<=0)
{
  perror("Error de lectura en archivo de entrada");
  exit(1);
};
    };
  vermem(nombre_salida=(char *) malloc(sizeof(char)*(strlen(argv[1])+5)));
  strcpy(nombre_salida, argv[1]);
  strcat(nombre_salida, "z.tex");
  if ((salida=fopen(nombre_salida, "wt"))==NULL)
    {
      perror("Error en archivo de salida");
      exit(1);
    };
  vermem(nombre_salida_log=(char *) malloc(sizeof(char)*(strlen(argv[1])+6)));
  strcpy(nombre_salida_log, argv[1]);
  strcat(nombre_salida_log, ".zlog");
  if ((salida_log=fopen(nombre_salida_log, "wt"))==NULL)
    {
      perror("Error en archivo de salida");
      exit(1);
    };
  vermem(nombre_salida_log2=(char *) malloc(sizeof(char)*(strlen(argv[1])+7)));
  strcpy(nombre_salida_log2, argv[1]);
```

```
  strcat(nombre_salida_log2, ".zlog2");
  if ((salida_log2=fopen(nombre_salida_log2, "wt"))==NULL)
    {
      perror("Error en archivo de salida");
      exit(1);
    };


  vermem(tr=(triang *) malloc(sizeof(triang)));

  vermem(z=(zeta_vector *) malloc(sizeof(zeta_vector)));
  vermem(z_cas=(indice_de_casilla**) malloc(sizeof(indice_de_casilla*)));
  vermem(zeta=(zeta_vector *) malloc((2*tamanyo)*sizeof(zeta_vector)));
  vermem(*zeta=(zeta_vector) malloc((2*tamanyo)*sizeof(casilla)));

  CONTADOR=0;
  LONG_FILA_TeX=9;
  while (!feof(entrada))
    {
      lee_triangulo(tr, tamanyo, entrada);
      CONTADOR++;
      fprintf(stderr, "CONTADOR=%ld\n", CONTADOR);
      escribe_bullets_uno(*tr, tamanyo, salida);
      verifica(fprintf(salida, "\n\n"));
      num_de_ceros=numero_de_ceros(*tr,tamanyo);
      vermem(*z=(zeta_vector)
     malloc((num_de_ceros+(*tr)[0][0].longitud)*sizeof(casilla)));


      num_de_alphas_nulas=iniciar_zetas(z, z_cas, *tr, tamanyo, tamanyo-1);
      lema_dos_uno(z, z_cas, zeta, *tr, tamanyo, tamanyo-1,
   num_de_alphas_nulas, salida);
      se_puede_despejar=1;
      while (!zetas_iguales(zeta, tamanyo-1) && se_puede_despejar)
{
  verifica(fprintf(salida, "\nHay que hacer"
   " substituciones.\n"));
  se_puede_despejar=
    despejar_una_variable_en_lista(z, z_cas, zeta, *tr,
   tamanyo, tamanyo-1,
   num_de_alphas_nulas, salida,
   salida_log, salida_log2);
};
      if (OVERFLOW)
{
  mainvl(tr, tamanyo, z_cas, salida, salida_log, salida_log2);
  free(*z);
  continue;
};
      if (se_puede_despejar)
{
  verifica(fprintf(salida, "\n!‘Hecho!\n\n"));
};
      verifica(fprintf(salida, "\n\\’Estos son los valores"
       " de las alphas nulas y las $z_i$:\n"
       "\\begin{eqnarray*}\n"));
      for (i=0; i<num_de_alphas_nulas; i++)
```

```
{
  verifica(fprintf(salida, "\\alpha_{%d,%d} &=&",
   (*z_cas)[i][0], (*z_cas)[i][1]));
  escribe_casilla_archivo_TeX(*(*z+i), "z", salida);
  if (i<num_de_alphas_nulas-1)
    {
      verifica(fprintf(salida, "\\\\"));
    }
  verifica(fprintf(salida, "\n"));
};
      verifica(fprintf(salida,
       "\\end{eqnarray*}\n\\begin{eqnarray*}\n"));
      for (i=0; i<(2*tamanyo); i++)
{
  verifica(fprintf(salida, "z_{%d}&=&", i+1));
  escribe_casilla_archivo_TeX(*(*zeta+i), "z", salida);
  if (i<(2*tamanyo)-1)
    {
      verifica(fprintf(salida, "\\\\"));
    };
  verifica(fprintf(salida, "\n"));
};
      verifica(fprintf(salida, "\\end{eqnarray*}\n\\clearpage\n\n"));
      free(*z_cas);
      free(*z);
      fflush(salida);
      fflush(salida_log);
      fflush(salida_log2);
    };
  free(*zeta);
  fclose(entrada);
  fclose(salida);
  fclose(salida_log);
  fclose(salida_log2);
  free(nombre_salida);
  free(nombre_salida_log);
  free(nombre_salida_log2);
  free(tr);
  free(z);
  free(z_cas);
  free(zeta);
  return(CONTADOR);
}
```

A.1.13. El archivo `zeta2.c`

```
/* Archivo zeta.c */
/* Contiene funciones para argumentar con los z_k en los teoremas */
/* sobre los $p$-grupos de clase maximal. */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "bullet.h"
```

```
extern long int CONTADOR;
extern long int LONG_FILA_TeX;
extern long int OVERFLOW, STACK_OVERFLOW;

int zetas_iguales(zeta_vector *z, int c_cero)
{
  int res, i;

  res=1;
  for (i=1; (i<c_cero)&&res; i++)
    {
      if (!son_casillas_iguales(**z, *(*z+i)))
{
  res=0;
};
    };
  return(res);
}

int numero_de_ceros(triang tr, int fila)
{
  int num_de_ceros, i, j;
  num_de_ceros=0;
  for (i=fila-1;i>=0;i--)
    {
      for (j=0;j<=(i/2);j++)
{
  if (es_cero(tr[i][j]))
    {
      num_de_ceros++;
    };
};
    };
  return(num_de_ceros);
}

int iniciar_zetas(zeta_vector *z, indice_de_casilla **z_cas, triang
  tr, int fila, int c_cero)
{
  int i,j,k,num_de_ceros,u;

  num_de_ceros=numero_de_ceros(tr,fila);
  for (i=0;i<num_de_ceros+tr[0][0].longitud;i++)
    {
      (*z+i)->longitud=fila+c_cero+1;
      for (u=0;u<fila+c_cero+1;u++)
{
  (*z+i)->coef[u].num=0;
  (*z+i)->coef[u].den=1;
};
    };
  vermem(*z_cas=(indice_de_casilla*)
 malloc((num_de_ceros+tr[0][0].longitud) *
sizeof(indice_de_casilla)));
  for (i=0;i<tr[0][0].longitud;i++)
    {
```

```
      for (u=0;u<i+c_cero+1;u++)
{
  ((*z+i)->coef)[i+u].num=(u%2) ? (-choose(i+c_cero,u)) :
    (choose(i+c_cero,u));
  ((*z+i)->coef)[i+u].den=1;
};
      (*z_cas)[i][0]=i+1;
      (*z_cas)[i][1]=i+1;
    };
  k=tr[0][0].longitud;
  for(i=fila-1;i>=0;i--)
    {
      for (j=0;j<=(i/2);j++)
{
  if (es_cero(tr[i][j]))
    {
      for (u=0;u<i+2-j+c_cero;u++)
{
  ((*z+k)->coef[j+u]).num=(u%2) ?
    (-choose(i-j+1+c_cero,u)) :
      (choose(i-j+1+c_cero,u));
  ((*z+k)->coef[j+1]).den=1;
};
      (*z_cas)[k][0]=j+1;
      (*z_cas)[k][1]=i+2-j;
      k++;
    };
};
    };
  return(num_de_ceros+tr[0][0].longitud);
}

/* Definicion de numero combinatorio */
unsigned long int choose(unsigned long int x1, unsigned long int x2)
{
  unsigned long int i, res, temp;

  STACK_OVERFLOW++;
  if (STACK_OVERFLOW > MAX_STACK_OVERFLOW)
    {
      fprintf(stderr,"Desbordamiento de pila en funci\363n choose.\n");
      exit(1);
    };
  if (x2>x1)
    {
      STACK_OVERFLOW--;
      return ((unsigned long) 0);
    }
  else
    {
      if (2*x2>x1)
{
  STACK_OVERFLOW--;
  return(choose(x1,x1-x2));
}
      else
{
```

```
  res=1;
  for (i=0;i<x2;i++)
    {
      temp=(x1-i)/(i+1);
      if (log_2_int(temp)+log_2_int(res) > 31)
{
  OVERFLOW=1;
  return(res);
};
      res*=temp;
    };
  STACK_OVERFLOW--;
  return(res);
};
    };
}

void aplicar_lema_dos_uno(int mat[], int long_mat, int a, int b)
{
  int i, mat_a, mat_b;

  mat_a=mat[a-1];
  mat_b=mat[b-1];

  if (mat_a>mat_b)
    {
      aplicar_lema_dos_uno(mat, long_mat, b, a);
    }
  else
    {
      if (mat_a<mat_b)
{
  for (i=0;i<long_mat;i++)
    {
      if (mat[i]==mat_a)
{
  mat[i]=mat_b;
};
    };
};
    }; /* no se hace nada si mat[a-1]==mat[b-1] */
}

void iniciar_lema_dos_uno(int mat[], int long_mat)
{
  int i;

  for (i=0; i<long_mat; i++)
    {
      mat[i]=i+1;
    };
}

void lema_dos_uno(zeta_vector *z, indice_de_casilla **z_cas,
  zeta_vector *zeta, triang tr, int fila, int c_cero,
  int num_de_alphas_nulas, FILE *salida)
{
```

```
  int *mat;
  int i, j, k, long_mat;

  long_mat=fila+c_cero+1;
  vermem(mat=(int *) malloc(long_mat*sizeof(int)));
  iniciar_lema_dos_uno(mat, fila+c_cero+1);

  for (i=0;i<fila;i++)
    {
      for (j=0;j<=(i/2);j++)
{
  if (!es_cero(tr[i][j]))
    {
      if (j < (i/2))
{
  if (!es_cero(tr[i][j+1]))
    {
      aplicar_lema_dos_uno(mat, long_mat, j+1, i-j+1);
    };
};
      if (i < fila-1)
{
  if (j < ((i+1)/2))
    {
      if (!es_cero(tr[i+1][j+1]))
{
  aplicar_lema_dos_uno(mat, long_mat, j+1,
       3+i+c_cero);
}
    };
  if (!es_cero(tr[i+1][j]))
    {
      aplicar_lema_dos_uno(mat, long_mat, i+2-j,
   3+i+c_cero);
      if (j<((i+1)/2)-1)
{
  if (es_cero(tr[i+1][j+1]))
    {
      if (!es_cero(tr[i+1][j+2]))
{
  if (!es_cero(tr[i][j+1]))
    {
      aplicar_lema_dos_uno(mat,
   long_mat,
   j+2,
   i+2-j);
    }
};
    };
};
    };
};
    };
};
    };

  for(i=0; i<num_de_alphas_nulas;i++)
```

```
    {
      substituye_rapido_casilla(*z+i,mat);
    };
  verifica(fprintf(salida, "Por aplicaci\\'on del \\LemaDosUno,"
   " obtenemos:\n"));
  verifica(fprintf(salida, "\\begin{eqnarray*}\n"));
  for (i=0;i<fila+c_cero+1;i++)
    {
      (*zeta+i)->longitud=fila+c_cero+1;
      for (k=0; k<fila+c_cero+1; k++)
{
  (*zeta+i)->coef[k].num=0;
  (*zeta+i)->coef[k].den=1;
};
      (*zeta+i)->coef[mat[i]-1].num=1;
      verifica(fprintf(salida, "z_{%d}&=&", i+1));
      escribe_casilla_archivo_TeX(*(*zeta+i), "z", salida);
      if (i<fila+c_cero)
{
  verifica(fprintf(salida, "\\\\"));
};
      verifica(fprintf(salida, "\n"));
    };
  verifica(fprintf(salida, "\\end{eqnarray*}\n\n\\begin{eqnarray*}\n"));
  for (k=0; k<num_de_alphas_nulas; k++)
    {
      verifica(fprintf(salida, "\\alpha_{%d,%d}&=&", (*z_cas)[k][0],
      (*z_cas)[k][1]));
      escribe_casilla_archivo_TeX(*(*z+k), "z", salida);
      if (k<num_de_alphas_nulas-1)
{
  verifica(fprintf(salida, "\\\\"));
};
      verifica(fprintf(salida, "\n"));
    };
  verifica(fprintf(salida, "\\end{eqnarray*}\n\n"));
  free(mat);
}
/* en la fila i, columna j, nos aparece el $\alpha_{j+1, i+2-j}$ */

void substituye_rapido_casilla(casilla *cas, int mat[])
{
  int i;

  for (i=0; i<cas->longitud; i++)
    {
      if (i!=mat[i]-1)
{
  cas->coef[mat[i]-1]=sum(cas->coef[mat[i]-1], cas->coef[i]);
  cas->coef[i].num=0;
  cas->coef[i].den=1;
};
    };
}

void debuga_casillas(zeta_vector zv, int numero_de_filas)
{
```

```
  int i;

  for(i=0; i<numero_de_filas; i++)
    {
      escribe_casilla(*(zv+i));
      printf(" - [%d]\n",i);
    };
}


int despejar_una_variable_en_lista(zeta_vector *z, indice_de_casilla
   **z_cas, zeta_vector *zeta, triang
   tr, int fila, int c_cero, int
   num_de_alphas_nulas, FILE *salida, FILE
   *salida_log, FILE *salida_log2)
{
  despeje des, des1;
  int i, posicion_en_lista;
  long int p, mp;

  posicion_en_lista=-1;
  mp=0;
  for (i=0;i<num_de_alphas_nulas;i++)
    {
      if (!es_cero(*(*z+i)))
  {
    des1=despejar_una_variable(*(*z+i));
    p=mayor_primo(des1.cas.coef[des1.pos].num);
    if (OVERFLOW)
      {
break;
      };
    if ((mp>p) || (posicion_en_lista<0))
      {
mp=p;
des=des1;
posicion_en_lista=i;
if (mp==1)
  {
    break;
  };
      };
  };
    };
  if (OVERFLOW)
    {
      fprintf(stderr, "Detectado desbordamiento entero.\n");
      verifica(fprintf(salida, "\n\n{\\Huge\\bf Desbordamiento entero, "
      "empezamos de nuevo.}\n"));
      return(0);
    };
  if (posicion_en_lista < 0)
    {
      /* Lo siento, todo son ceros */
      verifica(fprintf(salida, "\n\n{\\Huge\\bf Imposible"
       " seguir adelante $\\ldots$}\n"));
      escribe_salida(tr,fila,salida_log);
```

```
      verifica(fprintf(salida_log2, "%ld\n", CONTADOR));

      return(0);
    }
  else
    {
      substituir_una_variable_en_lista(z, z_cas, zeta, tr, fila,
       c_cero, num_de_alphas_nulas,
       posicion_en_lista, des,
       salida);
      if (OVERFLOW)
{
  fprintf(stderr, "Desbordamiento entero detectado.\n");
  return(0);
};
      return(1);
    };
}

void substituir_una_variable_en_lista(zeta_vector *z,
      indice_de_casilla **z_cas,
      zeta_vector *zeta, triang tr,
      int fila, int c_cero, int
      num_de_alphas_nulas, int num_de_var,
      despeje des, FILE *salida)
{
  zeta_vector zz;
  int i;
  long int el_mcd;
  int *anulado, nanulados;

  /* Primero obtenemos los indices para los cuales cambiamos de cero a */
  /* no cero al hacer la substitucion.  Efectuamos primero la */
  /* substitucion.*/

  vermem(anulado=(int *) malloc((fila+c_cero+1)*sizeof(int)));
  nanulados=0;
  vermem(zz=(zeta_vector) malloc(num_de_alphas_nulas*sizeof(casilla)));
  for (i=0; i<num_de_alphas_nulas;i++)
    {
      *(zz+i)=
substituir(*(*z+i),des);
      if (OVERFLOW)
{
  break;
};
      if(!es_cero(*(*z+i)))
{
  if (es_cero(*(zz+i)))
    {
      anulado[nanulados]=i;
      nanulados++;
    };
};
    };
  for (i=0;i<fila+c_cero+1;i++)
    {
```

```
/*      for (k=0; k<fila+c_cero+1; k++)
{
  escribe_casilla(*(*zeta+k));
  printf("=z_%d\n", k+1);
};
      for (k=0; k<num_de_ceros; k++)
{
  escribe_casilla(*(*z+k));
  printf("=\\alpha correspondiente a k=%d\n", k);
};
*/      if (!es_cero(*(*zeta+i)))
{
  *(*zeta+i)=
    substituir(*(*zeta+i),des);
  if (OVERFLOW)
    {
      break;
    };
};
    };
  if (OVERFLOW)
    {
      return;
    };
  if (!nanulados)
    {
      anulado[0]=0;
    };
  el_mcd=(*z+anulado[0])->coef[des.pos].num;
  if (nanulados==1)
    {
      verifica(fprintf(salida,"\nRealizamos una substituci\\'on,"
       " a partir del valor"
       " de $\\alpha_{%d,%d}$:\n",
       (*z_cas)[anulado[0]][0],
       (*z_cas)[anulado[0]][1]));
    }
  else
    {
      verifica(fprintf(salida, "\nRealizamos una substituci\\'on, a"
       " partir de los valores "
       "de $\\alpha_{%d,%d}$",
       (*z_cas)[anulado[0]][0],
       (*z_cas)[anulado[0]][1]));
      for (i=1;i<nanulados;i++)
{
  el_mcd=gcd(el_mcd, (*z+anulado[i])->coef[des.pos].num);
  if (OVERFLOW)
    {
      break;
    };
  if (i<nanulados-1)
    {
      verifica(fprintf(salida, ", "));
    }
  else
    {
```

```
      verifica(fprintf(salida, " y "));
    };
  verifica(fprintf(salida, "$\\alpha_{%d,%d}$",
   (*z_cas)[anulado[i]][0],
   (*z_cas)[anulado[i]][1]));
};
      verifica(fprintf(salida, ":\n"));
    };
  if (OVERFLOW)
    {
      return;
    };
  verifica(fprintf(salida, "\\begin{eqnarray*}\n"));
  for (i=0;i<nanulados;i++)
    {
      verifica(fprintf(salida, "\\alpha_{%d,%d} &=& ",
       (*z_cas)[anulado[i]][0],
       (*z_cas)[anulado[i]][1]));
      escribe_casilla_archivo_TeX(*(*z+(anulado[i])), "z", salida);
      if (i<nanulados-1)
{
  verifica(fprintf(salida, "\\\\"));
};
      verifica(fprintf(salida, "\n"));
    };
  verifica(fprintf(salida, "\\end{eqnarray*}\n$$z_{%d}=", des.pos+1));
  escribe_casilla_archivo_TeX(*(*zeta+des.pos), "z", salida);
  verifica(fprintf(salida, "$$\n"));
  verifica(fprintf(salida, "\nEsta substituci\\'on es v\\'alida para"
   " $p>%ld$.\n",
   mayor_primo(el_mcd)));
  for (i=0; i<num_de_alphas_nulas;i++)
    {
      *(*z+i)
=*(zz+i);
    };
  free(zz);
  free(anulado);
}
```

A.1.14. El archivo `gros.c`

```
#include <stdio.h>
#include "bullet.h"
#include "bulletvl.h"
long int CONTADOR, LONG_FILA_TeX, STACK_OVERFLOW, OVERFLOW;

verylong longint2vl(unsigned long int a)
{
  verylong res;
  int k;

  res.sgn=0;
  res.coef[0]=a&0xFFFF;
```

```
  res.coef[1]=a>>16;
  for (k=2;k<MAYOR;k++)
    {
      res.coef[k]=0;
    };
  return(res);
};


verylong sumavl(verylong a, verylong b)
{
  verylong res;
  int i;

  if (es_cerovl(a))
    {
      return(b);
    };
  if (es_cerovl(b))
    {
      return(a);
    };
  for (i=0;i<MAYOR;i++)
    {
      res.coef[i]=0;
    };
  res.sgn=0;
  if (a.sgn==b.sgn)
    {
      res.sgn=a.sgn;
      for(i=0;i<MAYOR-1;i++)
{
  res.coef[i]+=a.coef[i]+b.coef[i];
  if (res.coef[i] > 0xFFFF)
    {
      res.coef[i+1]++;
      res.coef[i] &= 0xFFFF;
    };
};
      res.coef[MAYOR-1]+=a.coef[MAYOR-1]+b.coef[MAYOR-1];
      if (res.coef[MAYOR-1] > 0xFFFF)
{
  fprintf(stderr, "Desbordamiento en funci\363n sumavl:\n");
  exit(1);
};
    };
  if (a.sgn!=b.sgn)
    {
      if (!a.sgn) /* a>0>b */
{
  if (es_mayorvl(a,menosvl(b)))
    {
      res=(restavl(a, menosvl(b)));
    }
  else
    {
      res=(menosvl(restavl(menosvl(b), a)));
```

```
    };
}
      else     /* b > 0 > a */
{
  if (es_mayorvl(b,menosvl(a)))
    {
      res=(restavl(b, menosvl(a)));
    }
  else
    {
      res=(menosvl(restavl(menosvl(a), b)));
    };
}
    }
/*  representa(a);printf("+"); representa(b);printf("=");
  representa(res); printf("\n");
*/  return(res);
};

void escribevl(verylong a)
{

  unsigned long int a_dec[2*MAYOR];
  verylong b;
  long int k, j;

  /* Lo que hacemos para escribir un entero "verylong" es pasarlo a */
  /* base 10000 y luego imprimirlo. */

  if (a.sgn==1)
    {
      printf("-");
    };
  b=a;
  k=0;
  while (!es_cerovl(b))
    {
      a_dec[k]=mod_intvl(b,10000);
      b=divvl(b,longint2vl(10000));
      k++;
    };
  k--;
  printf("%lu", a_dec[k]);
  for (j=k-1;j>=0;j--)
    {
      printf("%04lu",a_dec[j]);
    };
}

void escribevl_archivo(FILE *archivo, verylong a)
{

  unsigned long int a_dec[2*MAYOR];
  verylong b;
  long int k, j;

  /* Lo que hacemos para escribir un entero "verylong" es pasarlo a */
```

```
  /* base 10000 y luego imprimirlo. */

  if (a.sgn==1)
    {
      verifica(fprintf(archivo,"-"));
    };
  b=a;
  k=0;
  while (!es_cerovl(b))
    {
      a_dec[k]=mod_intvl(b,10000);
      b=divvl(b,longint2vl(10000));
      k++;
    };
  k--;
  verifica(fprintf(archivo,"%lu", a_dec[k]));
  for (j=k-1;j>=0;j--)
    {
      verifica(fprintf(archivo,"%04lu",a_dec[j]));
    };
}



verylong restavl(verylong a, verylong b)
{
  /* En esta funcion se supone que a>b>0 */
  verylong min, sub, res;
  int i;

  if (es_mayorvl(b,a))
    {
      return (menosvl(restavl(b,a)));
    };
  for (i=0; i<MAYOR; i++)
    {
      min.coef[i]=a.coef[i];
      sub.coef[i]=b.coef[i];
    };
  min.sgn=0;
  sub.sgn=0;

  res.sgn=0;
  for (i=0; i<MAYOR; i++)
    {
      if (min.coef[i]>=sub.coef[i])
{
  res.coef[i]=min.coef[i]-sub.coef[i];
}
      else
{
  min.coef[i]+=0x10000;
  sub.coef[i+1]+= 1;
  res.coef[i]=min.coef[i]-sub.coef[i];
};
    };
```

```
  return(res);
}

int es_cerovl(verylong a)
{
  int i;

  for (i=0;(i<MAYOR) && !(a.coef[i]); i++);
  return(i==MAYOR);
}

verylong absvl(verylong a)
{
  verylong res;
  int i;

  for (i=0;i<MAYOR; i++)
    {
      res.coef[i]=a.coef[i];
    };
  res.sgn=0;
  return(res);
}



int es_mayorvl(verylong a, verylong b)
{
  int i;
  if (a.sgn>b.sgn)
    {
      return(0);
    }
  if (a.sgn<b.sgn)
    {
      return(1);
    };
  if (a.sgn==1)
    {
      return(es_mayorvl(menosvl(b), menosvl(a)));
    };
  for (i=MAYOR-1; (i>=0) && (a.coef[i]==b.coef[i]); i--);
  if (i==-1)
    {
      return(0);
    }
  if (a.coef[i]<b.coef[i])
    {
      return(0);
    }
  else
    {
      return(1);
    };
};
```

```
int es_igualvl(verylong a, verylong b)
{
  int i, res;


  if (a.sgn!=b.sgn)
    {
      return(0);
    };
  res=1;
  for (i=0; (i<MAYOR) && (res) ;i++)
    {
      res=(a.coef[i]==b.coef[i]);
    };
  return(res);
}


verylong menosvl(verylong a)
{
  verylong res;
  int i;

  if (es_cerovl(a))
    {
      return(a);
    };
  for (i=0; i<MAYOR; i++)
    {
      res.coef[i]=a.coef[i];
    };
  res.sgn=1-a.sgn;
  return(res);
}

void representa(verylong a)
{
/*  int i;*/

/*  if (a.sgn)
    {
      printf("-");
    }
  for (i=MAYOR-1;i>=0; i--)
    {
      printf("%lu:", a.coef[i]);
    }
  printf("\n");*/
  escribevl(a);
/*  fflush(stdin);*/
};

verylong prodvl(verylong a, verylong b)
{
  int i, j;
  unsigned long int rcoef[MAYOR+1], parcial;
  verylong res;
```

```
  for (i=0; i<MAYOR+1; i++)
    {
      rcoef[i]=(unsigned long int) 0;
    };
  for (i=0; i<MAYOR; i++)
    {
      for (j=0; (j<=i)&& (j<MAYOR); j++)
{
  while ((i-j) > MAYOR)
    {
      j++;
    };
  parcial=a.coef[j]*b.coef[i-j];
  rcoef[i]+=parcial;
  if (parcial>rcoef[i])
    {
      rcoef[i+1]+=0x10000;
    };
};
      rcoef[i+1]+=rcoef[i] >> 16;
      rcoef[i] &= 0xFFFF;
    };
  if (rcoef[MAYOR]>0)
    {
      fprintf(stderr, "Desbordamiento en multiplicaci\363n.\n");
      exit(1);
    };
  res.sgn=(a.sgn!=b.sgn);
  for (i=0;i<MAYOR; i++)
    {
      res.coef[i]=rcoef[i];
    };
  return(res);
}

verylong div_intvl(verylong a, unsigned long int  b)
{
  verylong parcial, res;
  int i;

  if (!b)
    {
      fprintf(stderr, "Error: divisi\363n por cero\n");
      exit(1);
    };

  res.sgn=a.sgn;
  parcial.sgn=a.sgn;
  for (i=0; i<MAYOR; i++)
    {
      res.coef[i]=0;
      parcial.coef[i]=a.coef[i];
    };
  for (i=MAYOR-1; i>0; i--)
    {
      res.coef[i]=parcial.coef[i]/b;
```

```
      parcial.coef[i] %= b;
      parcial.coef[i-1] += 0x10000 * parcial.coef[i];
      parcial.coef[i]=0;
    };
  res.coef[0]=parcial.coef[0]/b;
  parcial.coef[0] %=b ;
  return (res);
}

unsigned long int mod_intvl(verylong a, unsigned long int  b)
{
  verylong parcial, res;
  int i;

  if (!b)
    {
      fprintf(stderr, "Error: divisi\363n por cero\n");
      exit(1);
    };

  if (b>0xFFFF)
    {
      fprintf(stderr, "Por favor, utilice las funciones asociadas "
      "a enteros largos.\n");
      exit(1);
    };

  res.sgn=a.sgn;
  parcial.sgn=a.sgn;
  for (i=0; i<MAYOR; i++)
    {
      res.coef[i]=0;
      parcial.coef[i]=a.coef[i];
    };
  for (i=MAYOR-1; i>0; i--)
    {
      res.coef[i]=parcial.coef[i]/b;
      parcial.coef[i] %= b;
      parcial.coef[i-1] += 0x10000 * parcial.coef[i];
      parcial.coef[i]=0;
    };
  res.coef[0]=parcial.coef[0]/b;
  parcial.coef[0] %=b ;
  return (parcial.coef[0]);
}
```

A.1.15. El archivo `fraccionvl.c`

```
#include <stdio.h>
#include <stdlib.h>
#include "bullet.h"
```

```
#include "bulletvl.h"

extern long int CONTADOR;
extern long int STACK_OVERFLOW;


/* La siguiente funcion suma dos fraccionvles. */
fraccionvl sumfrvl(fraccionvl primero, fraccionvl segundo)
{
  fraccionvl res, res2;

  res.num=sumavl(prodvl(primero.num, segundo.den), prodvl(primero.den,
  segundo.num));
  res.den=prodvl(primero.den, segundo.den);
  res2=simplificafrvl(res);
  return(res2);
}

/* Calculo del opuesto de una fraccionvl */

fraccionvl menosfrvl(fraccionvl fr)
{
  fraccionvl res;
  res.num=menosvl(fr.num);
  res.den=fr.den;
  return(simplificafrvl(res));
}


/* Funcion disenyada para simplificar una fraccionvl */

fraccionvl simplificafrvl(fraccionvl fr)
{
  verylong mcd;
  fraccionvl res;
  if (es_cerovl(fr.num))
    {
      res.num=longint2vl(0);
      res.den=longint2vl(1);
    }
  else
    {
      mcd=gcdvl(fr.num,fr.den);
      res.num=divvl(fr.num,mcd);
      res.den=divvl(fr.den,mcd);
    };
  if (res.den.sgn)
    {
      res.num.sgn=1-res.num.sgn;
      res.den.sgn=1-res.den.sgn;
    };
  return res;
}

fraccionvl prodfrvl(fraccionvl primero, fraccionvl segundo)
{
  fraccionvl t1, t2, t3, t4, t5, t6, res;
```

```
  t1=simplificafrvl(primero);
  t2=simplificafrvl(segundo);
  t3.num=t1.num;
  t3.den=t2.den;
  t4.num=t2.num;
  t4.den=t1.den;
  t5=simplificafrvl(t3);
  t6=simplificafrvl(t4);
  res.num=prodvl(t5.num,t6.num);
  res.den=prodvl(t5.den,t6.den);
  /*   res2=simplifica(res);*/
  return (res);
}

fraccionvl inversafrvl(fraccionvl fr)
{
  fraccionvl res;
  if (es_cerovl(fr.num))
    {
      fprintf ( stderr, "Error en funcion \"inversafrvl\": No puedo"
       "dividir por cero");
      exit(1);
    }
  else
    {
      res.num=fr.den;
      res.den=fr.num;
      return(res);
    };
}

verylong gcdvl(verylong a, verylong b)
{
  /* Nuestro objetivo aqui es aplicar el algoritmo de Euclides para el */
  /* calculo del mcd de dos numeros enteros.*/
  verylong res;

  STACK_OVERFLOW++;
  if (STACK_OVERFLOW > MAX_STACK_OVERFLOW)
    {
      fprintf(stderr,
      "Desbordamiento de pila en funcion gcdvl con parametros:\n");
      escribevl_archivo(stderr, a);
      fprintf(stderr, "\ny\n");
      escribevl_archivo(stderr, b);
      fprintf(stderr, "\n");
      exit(1);
    };
  if (es_cerovl(b))
    {
      STACK_OVERFLOW--;
      return (a);
    }
  else
    {
      if (b.sgn)
```

```
{
  res=gcdvl(a,menosvl(b));
  STACK_OVERFLOW--;
  return(res);
};
      if (a.sgn)
{
  res=gcdvl(menosvl(a),b);
  STACK_OVERFLOW--;
  return(res);
};
      res=gcdvl(b,modvl(a,b));
      STACK_OVERFLOW--;
      return(res);
    };

}

void escribe_fraccionvl(fraccionvl fr)
{
  printf("(");
  representa(fr.num);
  printf("/");
  representa(fr.den);
  printf(")");
}



verylong modvl(verylong a, verylong b)
{
  int na, nb;         /* ultimo "digito" significativo */

  verylong res;       /* el resultado que se debe devolver */
  long int  parcial[MAYOR+1];   /* resto parcial */

  int i, j, k;              /* contadores de bucle */

  int e;              /* un e de modo que multiplicaremos restos */
      /* parciales y divisor por 2^e */

  unsigned long int div1, div2;/* primera y segunda cifras, */
       /* respectivamente, del divisor, una */
       /* vez multiplicado por 2^e. */

  unsigned long int m01, m2; /* dos primeras cifras y tercera */
 /* cifra, respectivamente, de los */
 /* restos parciales.*/

  unsigned long int qi; /* cifra del cociente */
  unsigned long int c; /* producto de dos digitos */
  unsigned short int d; /* acarreo */

  /*Esta rutina ha sido adaptada a partir del programa GAP, archivo */
  /*"src/integer.c", por Martin Sch\"onert, Alice Niemeyer y Werner */
  /*Nickel, v 3.9, 1993/01/28 18:51:32, que por supuesto no tienen */
  /*nada que ver con los posibles desaguisados que puedan salir de */
```

```
  /*aqui. */

  /* Eliminemos, en primer lugar, el caso trivial en que el divisor es */
  /* cero. */

  if (es_cerovl(b))
    {
      fprintf(stderr,"Error: divisi\363n por cero.\n");
      exit(1);
    };


  /* Calculamos el numero de cifras significativas del dividendo y del */
  /* divisor.*/
  na=0;
  nb=0;
  for (i=0; i<MAYOR; i++)
    {
      if ((a.coef[i]))
{
  na=i;
};
      if ((b.coef[i]))
{
  nb=i;
};
    };

  /* El caso en que no hay cifras significativas en el divisor se */
  /* trata con una funcion especial */
  if (!nb)
    {
      res.sgn=a.sgn;
      for (k=1; k<MAYOR; k++)
{
  res.coef[k]=0;
};
      res.coef[0]=mod_intvl(a,b.coef[0]);
      return(res);
    };

  /* Tambien es trivial el caso en que hay mas cifras en el divisor */
  /* que en el dividendo. */
  if (nb>na)
    {
      return(a);
    };

  /* Los otros casos no han sido tratados anteriormente */

  /* Obtenemos primero un e adecuado tal que multiplicaremos los */
  /* restos parciales y el divisor por 2^e. */

  for (e=0; (b.coef[nb]<<e) + (b.coef[nb-1]>>(16-e)) < 0x8000; e++);
  div1= (b.coef[nb]<<e) + (b.coef[nb-1]>>(16-e));
  div2= ((b.coef[nb-1]<<e) + (nb>=2 ? b.coef[nb-2]>>(16-e) : 0))&0xFFFF;
```

```
  /* Ya tenemos el divisor multiplicado por un 2^e suficientemente */
  /* grande como para hacer los cocientes con un minimo de garantias. */
  /* Observese que solo hemos tomado dos cifras del divisor, ya que el */
  /* tipo long int tiene 32 bits.  A continuacion se van calculando */
  /* cifras del cociente y restos parciales.  */

  /* Antes de liarnos con el calculo de los cocientes y los restos */
  /* parciales, asignaremos el dividendo como primer resto parcial. */

  for (i=0; i<MAYOR; i++)
    {
      parcial[i]=a.coef[i];
    };
  parcial[MAYOR]=0;


  /* A continuacion nos queda devolver el resultado. */

  for (i=(na-nb); i>=0; i--)
    {
      /* primer paso: intentar adivinar el cociente parcial */
      j=i+nb;
      m01=((65536*parcial[j+1]+parcial[j])<<e) +
(parcial[j-1]>>(16-e));
      if (m01==0) continue; /*Si las dos  primeras cifras del resto */
    /*parcial son cero, "bajamos" la cifra */
    /*siguiente */
      m2=(parcial[j-1]<<e) + (j>=2 ? parcial[j-2]>>(16-e)
   : 0);
      /* Aqui se ha tratado de diferenciar los casos en que podemos */
      /* considerar o no cifras anteriores.  Hay que verificar la */
      /* condicion, que esta de prueba.*/

      if ((parcial[j+1]<<e) + (parcial[j]>>(16-e)) < div1)
{
  qi=m01 / div1; /* !‘tiene sentido la division entera! */
}
      else
{
  qi=0xFFFF; /* para las pruebas, tomaremos el mayor valor */
     /* posible*/
};
      while (m01-qi*div1<0x10000 && 0x10000 * (m01-qi*div1)+m2 < qi*div2)
{
  qi--;
};
      /* aqui se trata de afinar la cifra del cociente */

      /* a continuacion, se calcula el nuevo resto parcial */

      d=0;
      for (k=0; k<=nb; k++)
{
  c=parcial[k+i]-qi*b.coef[k]-d;
  parcial[k+i]=c&0xFFFF;
  d=-(c>>16);
}
```

```
      c=parcial[i+nb+1]-d;
      parcial[i+nb+1]=c&0xFFFF;
      d=-(c>>16);

      /* Si nos hemos quedado con valores finales negativos, anyadimos */
      /* de nuevo */
      if (d!=0)
{
  d=0;
  for (k=0; k<=nb; k++)
    {
      c=parcial[k+i]+b.coef[k]+d;
      parcial[k+i]=c &0xFFFF;
      d=(c>>16);
    };
  c=parcial[1+nb+i]+d;
  parcial[1+nb+i]=c& 0xFFFF;
  d=(c>>16);
  qi--;
};
      /* ?‘Funcionara? */
/*      printf("i=%i\nrestos parciales:", i);
      for (k=MAYOR-1;k>=0; k--)
{
  printf("%lu:",parcial[k]&0xFFFF);
};
      printf("\n");
*/
    };
  for (k=0;k<MAYOR; k++)
    {
      res.coef[k]=parcial[k];
    };
  res.sgn=a.sgn;

  return(res);


}



verylong divvl(verylong a, verylong b)
{
  int na, nb;         /* ultimo "digito" significativo */

  verylong res;       /* el resultado que se debe devolver */
  long int  parcial[MAYOR+1];   /* resto parcial */

  int i, j, k;              /* contadores de bucle */

  int e;              /* un e de modo que multiplicaremos restos */
      /* parciales y divisor por 2^e */

  unsigned long int div1, div2;/* primera y segunda cifras, */
       /* respectivamente, del divisor, una */
       /* vez multiplicado por 2^e. */
```

```
  unsigned long int m01, m2; /* dos primeras cifras y tercera */
 /* cifra, respectivamente, de los */
 /* restos parciales.*/

  unsigned long int qi; /* cifra del cociente */
  unsigned long int c; /* producto de dos digitos */
  unsigned short int d; /* acarreo */

  /*Esta rutina ha sido adaptada a partir del programa GAP, archivo */
  /*"src/integer.c", por Martin Sch\"onert, Alice Niemeyer y Werner */
  /*Nickel, v 3.9, 1993/01/28 18:51:32, que por supuesto no tienen */
  /*nada que ver con los posibles desaguisados que puedan salir de */
  /*aqui. */

  /* Eliminemos, en primer lugar, el caso trivial en que el divisor es */
  /* cero. */

  if (es_cerovl(b))
    {
      fprintf(stderr,"Error: divisi\363n por cero.\n");
      exit(1);
    };

  res.sgn=(a.sgn!=b.sgn);
  for (k=0; k<MAYOR; k++)
    {
      res.coef[k]=0;
    };


  /* Calculamos el numero de cifras significativas del dividendo y del */
  /* divisor.*/
  na=0;
  nb=0;
  for (i=0; i<MAYOR; i++)
    {
      if ((a.coef[i]))
{
  na=i;
};
      if ((b.coef[i]))
{
  nb=i;
};
    };

  /* El caso en que no hay cifras significativas en el divisor se */
  /* trata con una funcion especial */
  if (!nb)
    {
      return(div_intvl(a,b.coef[0]));
    };

  /* Tambien es trivial el caso en que hay mas cifras en el divisor */
  /* que en el dividendo. */
  if (nb>na)
```

```
    {
      res.sgn=0;
      for (i=0; i<MAYOR; i++)
{
  res.coef[i]=0;
};
      return(res);
    };

  /* Los otros casos no han sido tratados anteriormente */

  /* Obtenemos primero un e adecuado tal que multiplicaremos los */
  /* restos parciales y el divisor por 2^e. */

  for (e=0; (b.coef[nb]<<e) + (b.coef[nb-1]>>(16-e)) < 0x8000; e++);
  div1= (b.coef[nb]<<e) + (b.coef[nb-1]>>(16-e));
  div2= ((b.coef[nb-1]<<e) + (nb>=2 ? b.coef[nb-2]>>(16-e) : 0))&0xFFFF;

  /* Ya tenemos el divisor multiplicado por un 2^e suficientemente */
  /* grande como para hacer los cocientes con un minimo de garantias. */
  /* Observese que solo hemos tomado dos cifras del divisor, ya que el */
  /* tipo long int tiene 32 bits.  A continuacion se van calculando */
  /* cifras del cociente y restos parciales.  */

  /* Antes de liarnos con el calculo de los cocientes y los restos */
  /* parciales, asignaremos el dividendo como primer resto parcial. */

  for (i=0; i<MAYOR; i++)
    {
      parcial[i]=a.coef[i];
    };
  parcial[MAYOR]=0;

  for (i=(na-nb); i>=0; i--)
    {
      /* primer paso: intentar adivinar el cociente parcial */
      j=i+nb;
      m01=((65536*parcial[j+1]+parcial[j])<<e) +
(parcial[j-1]>>(16-e));
      if (m01==0) continue; /*Si las dos  primeras cifras del resto */
    /*parcial son cero, "bajamos" la cifra */
    /*siguiente */
      m2=(parcial[j-1]<<e) + (j>=2 ? parcial[j-2]>>(16-e)
   : 0);
      /* Aqui se ha tratado de diferenciar los casos en que podemos */
      /* considerar o no cifras anteriores.  Hay que verificar la */
      /* condicion, que esta de prueba.*/

      if ((parcial[j+1]<<e) + (parcial[j]>>(16-e)) < div1)
{
  qi=m01 / div1; /* !‘tiene sentido la division entera! */
}
      else
{
  qi=0xFFFF; /* para las pruebas, tomaremos el mayor valor */
     /* posible*/
};
```

```
      while (m01-qi*div1<0x10000 && 0x10000 * (m01-qi*div1)+m2 < qi*div2)
{
  qi--;
};
      /* aqui se trata de afinar la cifra del cociente */

      /* a continuacion, se calcula el nuevo resto parcial */

      d=0;
      for (k=0; k<=nb; k++)
{
  c=parcial[k+i]-qi*b.coef[k]-d;
  parcial[k+i]=c&0xFFFF;
  d=-(c>>16);
}
      c=parcial[i+nb+1]-d;
      parcial[i+nb+1]=c&0xFFFF;
      d=-(c>>16);

      /* Si nos hemos quedado con valores finales negativos, anyadimos */
      /* de nuevo */
      if (d!=0)
{
  d=0;
  for (k=0; k<=nb; k++)
    {
      c=parcial[k+i]+b.coef[k]+d;
      parcial[k+i]=c &0xFFFF;
      d=(c>>16);
    };
  c=parcial[1+nb+i]+d;
  parcial[1+nb+i]=c& 0xFFFF;
  d=(c>>16);
  qi--;
};
      /* ?‘Funcionara? */
      res.coef[i]=qi;
/*      printf("i=%i\nrestos parciales:", i);
      for (k=MAYOR-1;k>=0; k--)
{
  printf("%lu:",parcial[k]&0xFFFF);
};
      printf("\n");
*/
    };

  /* A continuacion nos queda devolver el resultado. */
  return(res);


}
```

A.1.16. El archivo `casillavl.c`

```
/* Archivo casillavl.c */
/* Operaciones con casillavls del triangulo T_G usando enteros */
/* larguisimos.*/

#include <stdio.h>
#include <stdlib.h>
#include "bullet.h"
#include "bulletvl.h"

casillavl prodescvl(fraccionvl escalar, casillavl cas)
{
  unsigned char i;
  casillavl res;
  res.longitud=cas.longitud;
  for (i=0; i<cas.longitud; i++)
    {
      res.coef[i]=prodfrvl(escalar, cas.coef[i]);
    };
  return(res);
}

casillavl menos_casvl(casillavl cas)
{
  fraccionvl menosuno;
  menosuno.num=longint2vl(1);
  menosuno.num.sgn=1;
  menosuno.den=longint2vl(1);
  return(prodescvl(menosuno,cas));
}

casillavl sum_casvl(casillavl cas1, casillavl cas2)
{
  unsigned char i;
  casillavl res;
  if (cas1.longitud!=cas2.longitud)
    {
      fprintf(stderr, "Error, solo esta definida la suma de"
      "\"casillavls\" de igual longitud");
      exit(1);
    }
  else
    {
      res.longitud=cas1.longitud;
      for (i=0;i<cas1.longitud;i++)
{
  res.coef[i]=sumfrvl(cas1.coef[i],cas2.coef[i]);
}
      return (res);
    };
}


int es_cero_casillavl(casillavl cas)
{
  int escero;
```

```
  int i;
  escero=1;
  for (i=0;
       i<cas.longitud;
       i++)
    {
      if (escero)
{
  escero=(es_cerovl(cas.coef[i].num));
};
    };
  return(escero);
}



void escribe_casillavl(casillavl c)
{
  int i;
  for (i=0;i<c.longitud;i++)
    {
      escribe_fraccionvl(c.coef[i]);
    };
  printf("\n");
}

int son_casillavls_iguales(casillavl cas1, casillavl cas2)
{
  int i, res;
  if (cas1.longitud!=cas2.longitud)
    {
      return 0;
    }
  else
    {
      res=1;
      for (i=0; (i<cas1.longitud) && res; i++)
{
  res= (es_igualvl(cas1.coef[i].num,cas2.coef[i].num) &&
es_igualvl(cas1.coef[i].den,cas2.coef[i].den));
};
      return(res);
    };
}
```

A.1.17. El archivo `zetavl.c`

```
/* Archivo zeta.c */
/* Contiene funciones para argumentar con los z_k en los teoremas */
/* sobre los $p$-grupos de clase maximal. */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#include "bullet.h"
#include "bulletvl.h"

long int CONTADOR;
long int LONG_FILA_TeX;
extern long int STACK_OVERFLOW;
extern long int OVERFLOW;

int zetas_igualesvl(zeta_vectorvl *z, int c_cero)
{
  int res, i;

  res=1;
  for (i=1; (i<c_cero)&&res; i++)
    {
      if (!son_casillavls_iguales(**z, *(*z+i)))
{
  res=0;
};
    };
  return(res);
}

/*int numero_de_ceros(triang tr, int fila)
{
  int num_de_ceros, i, j;
  num_de_ceros=0;
  for (i=fila-1;i>=0;i--)
    {
      for (j=0;j<=(i/2);j++)
{
  if (es_cero(tr[i][j]))
    {
      num_de_ceros++;
    };
};
    };
  return(num_de_ceros);
}
*/
int iniciar_zetasvl(zeta_vectorvl *z, indice_de_casilla **z_cas, triang
  tr, int fila, int c_cero)
{
  int i,j,k,num_de_ceros,u;

  num_de_ceros=numero_de_ceros(tr,fila);
  for (i=0;i<num_de_ceros+tr[0][0].longitud;i++)
    {
      (*z+i)->longitud=fila+c_cero+1;
      for (u=0;u<fila+c_cero+1;u++)
{
  (*z+i)->coef[u].num=longint2vl(0);
  (*z+i)->coef[u].den=longint2vl(1);
};
    };
  vermem(*z_cas=(indice_de_casilla*)
 malloc((num_de_ceros+tr[0][0].longitud) *
```

```
sizeof(indice_de_casilla)));
  for (i=0;i<tr[0][0].longitud;i++)
    {
      for (u=0;u<i+c_cero+1;u++)
{
  ((*z+i)->coef)[i+u].num=(u%2) ?
    menosvl(longint2vl((choose(i+c_cero,u)))) :
      longint2vl((choose(i+c_cero,u)));
  ((*z+i)->coef)[i+u].den=longint2vl(1);
};
      (*z_cas)[i][0]=i+1;
      (*z_cas)[i][1]=i+1;
    };
  k=tr[0][0].longitud;
  for(i=fila-1;i>=0;i--)
    {
      for (j=0;j<=(i/2);j++)
{
  if (es_cero(tr[i][j]))
    {
      for (u=0;u<i+2-j+c_cero;u++)
{
  ((*z+k)->coef[j+u]).num=(u%2) ?
    menosvl(longint2vl(choose(i-j+1+c_cero,u))) :
      longint2vl((choose(i-j+1+c_cero,u)));
  ((*z+k)->coef[j+1]).den=longint2vl(1);
};
      (*z_cas)[k][0]=j+1;
      (*z_cas)[k][1]=i+2-j;
      k++;
    };
};
    };
  return(num_de_ceros+tr[0][0].longitud);
}

/* Definicion de numero combinatorio */
/* unsigned long int choose(unsigned long int x1, unsigned long int x2)
{
  unsigned long int i, res;

  if (x2>x1)
    {
      return ((unsigned long) 0);
    }
  else
    {
      if (2*x2>x1)
{
  return(choose(x1,x1-x2));
}
      else
{
  res=1;
  for (i=0;i<x2;i++)
    {
      res=res*(x1-i)/(i+1);
```

```
    };
  return(res);
};
    };
}
*/
/*void aplicar_lema_dos_uno(int mat[], int long_mat, int a, int b)
{
  int i, mat_a, mat_b;

  mat_a=mat[a-1];
  mat_b=mat[b-1];

  if (mat_a>mat_b)
    {
      aplicar_lema_dos_uno(mat, long_mat, b, a);
    }
  else
    {
      if (mat_a<mat_b)
{
  for (i=0;i<long_mat;i++)
    {
      if (mat[i]==mat_a)
{
  mat[i]=mat_b;
};
    };
};
    }; */ /* no se hace nada si mat[a-1]==mat[b-1] */
/*
}

void iniciar_lema_dos_uno(int mat[], int long_mat)
{
  int i;

  for (i=0; i<long_mat; i++)
    {
      mat[i]=i+1;
    };
}
*/
void lema_dos_unovl(zeta_vectorvl *z, indice_de_casilla **z_cas,
  zeta_vectorvl *zeta, triang tr, int fila, int c_cero,
  int num_de_alphas_nulas, FILE *salida)
{
  int *mat;
  int i, j, k, long_mat;

  long_mat=fila+c_cero+1;
  vermem(mat=(int *) malloc(long_mat*sizeof(int)));
  iniciar_lema_dos_uno(mat, fila+c_cero+1);

  for (i=0;i<fila;i++)
    {
      for (j=0;j<=(i/2);j++)
```

```
{
  if (!es_cero(tr[i][j]))
    {
      if (j < (i/2))
{
  if (!es_cero(tr[i][j+1]))
    {
      aplicar_lema_dos_uno(mat, long_mat, j+1, i-j+1);
    };
};
      if (i < fila-1)
{
  if (j < ((i+1)/2))
    {
      if (!es_cero(tr[i+1][j+1]))
{
  aplicar_lema_dos_uno(mat, long_mat, j+1,
       3+i+c_cero);
}
    };
  if (!es_cero(tr[i+1][j]))
    {
      aplicar_lema_dos_uno(mat, long_mat, i+2-j,
   3+i+c_cero);
      if (j<((i+1)/2)-1)
{
  if (es_cero(tr[i+1][j+1]))
    {
      if (!es_cero(tr[i+1][j+2]))
{
  if (!es_cero(tr[i][j+1]))
    {
      aplicar_lema_dos_uno(mat,
   long_mat,
   j+2,
   i+2-j);
    }
};
    };
};
    };
};
    };
};
    };

  for(i=0; i<num_de_alphas_nulas;i++)
    {
      substituye_rapido_casillavl(*z+i,mat);
    };
  verifica(fprintf(salida, "Por aplicaci\\'on del \\LemaDosUno,"
   " obtenemos:\n"));
  verifica(fprintf(salida, "\\begin{eqnarray*}\n"));
  for (i=0;i<fila+c_cero+1;i++)
    {
      (*zeta+i)->longitud=fila+c_cero+1;
      for (k=0; k<fila+c_cero+1; k++)
```

```
{
  (*zeta+i)->coef[k].num=longint2vl(0);
  (*zeta+i)->coef[k].den=longint2vl(1);
};
      (*zeta+i)->coef[mat[i]-1].num=longint2vl(1);
      verifica(fprintf(salida, "z_{%d}&=&", i+1));
      escribe_casillavl_archivo_TeX(*(*zeta+i), "z", salida);
      if (i<fila+c_cero)
{
  verifica(fprintf(salida, "\\\\"));
};
      verifica(fprintf(salida, "\n"));
    };
  verifica(fprintf(salida, "\\end{eqnarray*}\n\n\\begin{eqnarray*}\n"));
  for (k=0; k<num_de_alphas_nulas; k++)
    {
      verifica(fprintf(salida, "\\alpha_{%d,%d}&=&", (*z_cas)[k][0],
      (*z_cas)[k][1]));
      escribe_casillavl_archivo_TeX(*(*z+k), "z", salida);
      if (k<num_de_alphas_nulas-1)
{
  verifica(fprintf(salida, "\\\\"));
};
      verifica(fprintf(salida, "\n"));
    };
  verifica(fprintf(salida, "\\end{eqnarray*}\n\n"));
  free(mat);
}
/* en la fila i, columna j, nos aparece el $\alpha_{j+1, i+2-j}$ */

void substituye_rapido_casillavl(casillavl *cas, int mat[])
{
  int i;

  for (i=0; i<cas->longitud; i++)
    {
      if (i!=mat[i]-1)
{
  cas->coef[mat[i]-1]=sumfrvl(cas->coef[mat[i]-1], cas->coef[i]);
  cas->coef[i].num=longint2vl(0);
  cas->coef[i].den=longint2vl(1);
};
    };
}

void debuga_casillavls(zeta_vectorvl zv, int numero_de_filas)
{
  int i;

  for(i=0; i<numero_de_filas; i++)
    {
      escribe_casillavl(*(zv+i));
      printf(" - [%d]\n",i);
    };
}
```

```
int despejar_una_variablevl_en_lista(zeta_vectorvl *z, indice_de_casilla
   **z_cas, zeta_vectorvl *zeta, triang
   tr, int fila, int c_cero, int
   num_de_alphas_nulas, FILE *salida, FILE
   *salida_log, FILE *salida_log2)
{
  despejevl des, des1;
  int i, posicion_en_lista;
  verylong  p, mp;

  posicion_en_lista=-1;
  mp=longint2vl(0);
  for (i=0;i<num_de_alphas_nulas;i++)
    {
      if (!es_cero_casillavl(*(*z+i)))
  {
    des1=despejar_una_variablevl(*(*z+i));
    p=mayor_primovl(des1.cas.coef[des1.pos].num,(es_cerovl(mp)
 ? MAXPR :
 mp));
    if (es_mayorvl(mp,p) || (posicion_en_lista<0))
      {
mp=p;
des=des1;
posicion_en_lista=i;
if (es_igualvl(mp,longint2vl(1)))
  {
    break;
  };
      };
  };
    };
  if (posicion_en_lista < 0)
    {
      /* Lo siento, todo son ceros */
      verifica(fprintf(salida, "\n\n{\\Huge\\bf Imposible"
       " seguir adelante $\\ldots$}\n"));
      escribe_salida(tr,fila,salida_log);
      verifica(fprintf(salida_log2, "%ld\n", CONTADOR));

      return(0);
    }
  else
    {
      substituir_una_variablevl_en_lista(z, z_cas, zeta, tr, fila,
       c_cero, num_de_alphas_nulas,
       posicion_en_lista, des,
       salida);
      return(1);
    };
}

void substituir_una_variablevl_en_lista(zeta_vectorvl *z,
      indice_de_casilla **z_cas,
      zeta_vectorvl *zeta, triang tr,
      int fila, int c_cero, int
      num_de_alphas_nulas, int num_de_var,
```

```
      despejevl des, FILE *salida)
{
  zeta_vectorvl zz;
  int i;
  verylong el_mcd;
  int *anulado, nanulados;

  /* Primero obtenemos los indices para los cuales cambiamos de cero a */
  /* no cero al hacer la substitucion.  Efectuamos primero la */
  /* substitucion.*/

  vermem(anulado=(int *) malloc((fila+c_cero+1)*sizeof(int)));
  nanulados=0;
  vermem(zz=(zeta_vectorvl) malloc(num_de_alphas_nulas*sizeof(casillavl)));
  for (i=0; i<num_de_alphas_nulas;i++)
    {
      *(zz+i)=
substituirvl(*(*z+i),des);
      if(!es_cero_casillavl(*(*z+i)))
{
  if (es_cero_casillavl(*(zz+i)))
    {
      anulado[nanulados]=i;
      nanulados++;
    };
};
    };
  for (i=0;i<fila+c_cero+1;i++)
    {
      if (!es_cero_casillavl(*(*zeta+i)))
{
  *(*zeta+i)=
    substituirvl(*(*zeta+i),des);
};
    };
  if (!nanulados)
    {
      anulado[0]=0;
    };
  el_mcd=(*z+anulado[0])->coef[des.pos].num;
      printf("Substitucion de z_%d.\n\t a partir de la fila %d\n",
     des.pos+1, anulado[0]);
  if (nanulados==1)
    {
      verifica(fprintf(salida,"\nRealizamos una substituci\\'on,"
       " a partir del valor"
       " de $\\alpha_{%d,%d}$:\n",
       (*z_cas)[anulado[0]][0],
       (*z_cas)[anulado[0]][1]));
    }
  else
    {
      verifica(fprintf(salida, "\nRealizamos una substituci\\'on, a"
       " partir de los valores "
       "de $\\alpha_{%d,%d}$",
       (*z_cas)[anulado[0]][0],
       (*z_cas)[anulado[0]][1]));
```

```
      for (i=1;i<nanulados;i++)
{
  el_mcd=gcdvl(el_mcd, (*z+anulado[i])->coef[des.pos].num);
  if (i<nanulados-1)
    {
      verifica(fprintf(salida, ", "));
    }
  else
    {
      verifica(fprintf(salida, " y "));
    };
  verifica(fprintf(salida, "$\\alpha_{%d,%d}$",
   (*z_cas)[anulado[i]][0],
   (*z_cas)[anulado[i]][1]));
};
      verifica(fprintf(salida, ":\n"));
    };
  verifica(fprintf(salida, "\\begin{eqnarray*}\n"));
  for (i=0;i<nanulados;i++)
    {
      verifica(fprintf(salida, "\\alpha_{%d,%d} &=& ",
       (*z_cas)[anulado[i]][0],
       (*z_cas)[anulado[i]][1]));
      escribe_casillavl_archivo_TeX(*(*z+(anulado[i])), "z", salida);
      if (i<nanulados-1)
{
  verifica(fprintf(salida, "\\\\"));
};
      verifica(fprintf(salida, "\n"));
    };
  verifica(fprintf(salida, "\\end{eqnarray*}\n$$z_{%d}=", des.pos+1));
  escribe_casillavl_archivo_TeX(*(*zeta+des.pos), "z", salida);
  verifica(fprintf(salida, "$$\n"));
  verifica(fprintf(salida, "\nEsta substituci\\'on es v\\'alida para"
   " $p>%ld$.\n",
   mayor_primovl(el_mcd, MAXPR).coef[0] + 65536 *
   mayor_primovl(el_mcd, MAXPR).coef[1]));
  for (i=0; i<num_de_alphas_nulas;i++)
    {
      *(*z+i)
=*(zz+i);
    };
  printf("Nuevas casillas:\n");
  debuga_casillavls(zz, 18);
  printf("\n");
  free(zz);
  free(anulado);
}

int mainvl(triang *tr, int tamanyo, indice_de_casilla **z_cas, FILE
   *salida, FILE*salida_log, FILE*salida_log2)
{
  int se_puede_despejar;
  zeta_vectorvl *zvl, *zetavl;
  int num_de_alphas_nulas, i, num_de_ceros;

  fprintf(stderr, "CONTADOR=%ld - Enteros muy grandes...\n", CONTADOR);
```

```
/*  escribe_bullets_uno(*tr, tamanyo, salida);*/
  verifica(fprintf(salida, "\n\nRepetimos con enteros largos.\n\n"));
  num_de_ceros=numero_de_ceros(*tr,tamanyo);
  vermem(zvl=(zeta_vectorvl*)
 malloc(sizeof(zeta_vector)));
  vermem(*zvl=(zeta_vectorvl)
 malloc((num_de_ceros+(*tr)[0][0].longitud)*sizeof(casillavl)));
  vermem(zetavl=(zeta_vectorvl*)
 malloc(sizeof(zeta_vector)));
  vermem(*zetavl=(zeta_vectorvl)
 malloc((2*tamanyo)*sizeof(casillavl)));


  num_de_alphas_nulas=iniciar_zetasvl(zvl, z_cas, *tr, tamanyo, tamanyo-1);
  lema_dos_unovl(zvl, z_cas, zetavl, *tr, tamanyo, tamanyo-1,
 num_de_alphas_nulas, salida);
  se_puede_despejar=1;
  while (!zetas_igualesvl(zetavl, tamanyo-1) && se_puede_despejar)
    {
      verifica(fprintf(salida, "\nHay que hacer"
       " substituciones.\n"));
      se_puede_despejar=
despejar_una_variablevl_en_lista(zvl, z_cas, zetavl, *tr,
 tamanyo, tamanyo-1,
 num_de_alphas_nulas, salida,
 salida_log, salida_log2);
    };
  if (se_puede_despejar)
    {
      verifica(fprintf(salida, "\n!‘Hecho!\n\n"));
    };
  verifica(fprintf(salida, "\n\\’Estos son los valores"
   " de las alphas nulas y las $z_i$:\n"
   "\\begin{eqnarray*}\n"));
  for (i=0; i<num_de_alphas_nulas; i++)
    {
      verifica(fprintf(salida, "\\alpha_{%d,%d} &=&",
       (*z_cas)[i][0], (*z_cas)[i][1]));
      escribe_casillavl_archivo_TeX(*(*zvl+i), "z", salida);
      if (i<num_de_alphas_nulas-1)
{
  verifica(fprintf(salida, "\\\\"));
}
      verifica(fprintf(salida, "\n"));
    };
  verifica(fprintf(salida,
   "\\end{eqnarray*}\n\\begin{eqnarray*}\n"));
  for (i=0; i<(2*tamanyo); i++)
    {
      verifica(fprintf(salida, "z_{%d}&=&", i+1));
      escribe_casillavl_archivo_TeX(*(*zetavl+i), "z", salida);
      if (i<(2*tamanyo)-1)
{
  verifica(fprintf(salida, "\\\\"));
};
      verifica(fprintf(salida, "\n"));
    };
```

```
  verifica(fprintf(salida, "\\end{eqnarray*}\n\\clearpage\n\n"));
  free(*z_cas);
  free(*zetavl);
  free(zetavl);
  free(*zvl);
  free(zvl);
  fflush(salida);
  fflush(salida_log);
  fflush(salida_log2);
  return (CONTADOR);
}


int main_esp(int argc, char **argv)
{
  int se_puede_despejar;
  zeta_vector *z, *zeta;
  indice_de_casilla **z_cas;
  triang *tr;
  int tamanyo, num_de_alphas_nulas, i, num_de_ceros;
  FILE *entrada, *salida, *salida_log, *salida_log2;
  char *nombre_salida, *nombre_salida_log, *nombre_salida_log2;

  if ((entrada=fopen(argv[1], "rt"))==NULL)
    {
      perror("Error en archivo de entrada");
    }
  else
    {
      if (fscanf(entrada, "%d ", &tamanyo)<=0)
{
  perror("Error de lectura en archivo de entrada");
  exit(1);
};
    };
  vermem(nombre_salida=(char *) malloc(sizeof(char)*(strlen(argv[1])+5)));
  strcpy(nombre_salida, argv[1]);
  strcat(nombre_salida, "z.tex");
  if ((salida=fopen(nombre_salida, "wt"))==NULL)
    {
      perror("Error en archivo de salida");
      exit(1);
    };
  vermem(nombre_salida_log=(char *) malloc(sizeof(char)*(strlen(argv[1])+6)));
  strcpy(nombre_salida_log, argv[1]);
  strcat(nombre_salida_log, ".zlog");
  if ((salida_log=fopen(nombre_salida_log, "wt"))==NULL)
    {
      perror("Error en archivo de salida");
      exit(1);
    };
  vermem(nombre_salida_log2=(char *) malloc(sizeof(char)*(strlen(argv[1])+7)));
  strcpy(nombre_salida_log2, argv[1]);
  strcat(nombre_salida_log2, ".zlog2");
  if ((salida_log2=fopen(nombre_salida_log2, "wt"))==NULL)
    {
      perror("Error en archivo de salida");
```

```
      exit(1);
    };


  vermem(tr=(triang *) malloc(sizeof(triang)));

  vermem(z=(zeta_vector *) malloc(sizeof(zeta_vector)));
  vermem(z_cas=(indice_de_casilla**) malloc(sizeof(indice_de_casilla*)));
  vermem(zeta=(zeta_vector *) malloc((2*tamanyo)*sizeof(zeta_vector)));
  vermem(*zeta=(zeta_vector) malloc((2*tamanyo)*sizeof(casilla)));

  CONTADOR=0;
  LONG_FILA_TeX=9;
  while (!feof(entrada))
    {
      lee_triangulo(tr, tamanyo, entrada);
      OVERFLOW=0;
      CONTADOR++;
      fprintf(stderr, "CONTADOR=%ld\n", CONTADOR);
      escribe_bullets_uno(*tr, tamanyo, salida);
      verifica(fprintf(salida, "\n\n"));
      num_de_ceros=numero_de_ceros(*tr,tamanyo);
      vermem(*z=(zeta_vector)
     malloc((num_de_ceros+(*tr)[0][0].longitud)*sizeof(casilla)));


      num_de_alphas_nulas=iniciar_zetas(z, z_cas, *tr, tamanyo, tamanyo-1);
      lema_dos_uno(z, z_cas, zeta, *tr, tamanyo, tamanyo-1,
   num_de_alphas_nulas, salida);
      se_puede_despejar=1;
      while (!zetas_iguales(zeta, tamanyo-1) && se_puede_despejar)
{
  verifica(fprintf(salida, "\nHay que hacer"
   " substituciones.\n"));
  se_puede_despejar=
    despejar_una_variable_en_lista(z, z_cas, zeta, *tr,
   tamanyo, tamanyo-1,
   num_de_alphas_nulas, salida,
   salida_log, salida_log2);
};
      if (se_puede_despejar)
{
  verifica(fprintf(salida, "\n!‘Hecho!\n\n"));
};
      verifica(fprintf(salida, "\n\\’Estos son los valores"
       " de las alphas nulas y las $z_i$:\n"
       "\\begin{eqnarray*}\n"));
      for (i=0; i<num_de_alphas_nulas; i++)
{
  verifica(fprintf(salida, "\\alpha_{%d,%d} &=&",
   (*z_cas)[i][0], (*z_cas)[i][1]));
  escribe_casilla_archivo_TeX(*(*z+i), "z", salida);
  if (i<num_de_alphas_nulas-1)
    {
      verifica(fprintf(salida, "\\\\"));
    }
  verifica(fprintf(salida, "\n"));
```

```
};
      verifica(fprintf(salida,
       "\\end{eqnarray*}\n\\begin{eqnarray*}\n"));
      for (i=0; i<(2*tamanyo); i++)
{
  verifica(fprintf(salida, "z_{%d}&=&", i+1));
  escribe_casilla_archivo_TeX(*(*zeta+i), "z", salida);
  if (i<(2*tamanyo)-1)
    {
      verifica(fprintf(salida, "\\\\"));
    };
  verifica(fprintf(salida, "\n"));
};
      verifica(fprintf(salida, "\\end{eqnarray*}\n\\clearpage\n\n"));
      free(*z_cas);
      free(*z);
      fflush(salida);
      fflush(salida_log);
      fflush(salida_log2);
    };
  free(*zeta);
  fclose(entrada);
  fclose(salida);
  fclose(salida_log);
  fclose(salida_log2);
  free(nombre_salida);
  free(nombre_salida_log);
  free(nombre_salida_log2);
  free(tr);
  free(z);
  free(z_cas);
  free(zeta);
  return(CONTADOR);
}
```

A.1.18. El archivo `outputvl.c`

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
#include "bullet.h"
#include "bulletvl.h"

extern long int CONTADOR;

/* void verifica(int valor)
{
  if (valor<0)
    {
      perror("Error de escritura");
      exit(1);
    };
}
```

```
void lee_triangulo(triang *tr, int fila, FILE *archivo)
{
  int i,j, longitud;

  longitud=(fila+1)/2;
  for (i=fila-1;i>=0;i--)
    {
      for (j=0;j<=(i/2);j++)
{
  lee_casilla(&((*tr)[i][j]), longitud, archivo);
};
    };
}
*/
/*void lee_casillavl(casillavl *cas, int longitud, FILE *archivo)
{
  int j;

  cas->longitud=longitud;
  for (j=0; j<longitud; j++)
    {
      if (!fscanf(archivo,"(%ld/%ld)", &(cas->coef[j].num), &(cas->coef[j].den)))
{
  printf ("Error %c\n", '\007');
};
    };
  fscanf(archivo," ");
}
*/

void escribe_fraccionvl_archivo(FILE *archivo, fraccionvl fr)
{
  verifica(fprintf(archivo, "("));
  escribevl_archivo(archivo, fr.num);
  verifica(fprintf(archivo, "/"));
  escribevl_archivo(archivo, fr.den);
  verifica(fprintf(archivo, ")"));
};

void escribe_casillavl_archivo(casillavl cas, int longitud, FILE *archivo)
{
  int j;

  cas.longitud=longitud;
  for (j=0; j<longitud; j++)
    {
      escribe_fraccionvl_archivo(archivo,cas.coef[j]);
    };
  verifica(fprintf(archivo," "));
}

/*void escribe_triangulo_archivo(triang tr, int fila, FILE *archivo)
{
  int i,j, longitud;

  longitud=(fila+1)/2;
```

```
  for (i=fila-1;i>=0;i--)
    {
      for (j=0;j<=(i/2);j++)
{
  escribe_casilla_archivo(tr[i][j], longitud, archivo);
};
      verifica(fprintf(archivo,"\n"));
    };
  verifica(fprintf(archivo,"\n"));
}
*/

void escribe_casillavl_archivo_TeX(casillavl cas, char *nom_var, FILE
 *archivo)
{
  int j;
  int salio_primer_elemento;

  if (es_cero_casillavl(cas))
    {
      verifica(fprintf(archivo, "0"));
    }
  else
    {
      salio_primer_elemento=0;
      for (j=0; j<cas.longitud; j++)
{
  if (!es_cerovl(cas.coef[j].num))
    {
      if (!es_igualvl(cas.coef[j].den,longint2vl(1)))
{
  if (!es_igualvl(cas.coef[j].num,longint2vl(0)) &&
      !cas.coef[j].num.sgn)
    {
      if (!salio_primer_elemento)
{
  escribe_fraccionvl_archivo(archivo, cas.coef[j]);
  verifica(fprintf(archivo,"%s_{%d}", nom_var,
   j+1));
  salio_primer_elemento=1;
}
      else /* ya salio primer elemento */
{
  verifica(fprintf(archivo, "+"));
  escribe_fraccionvl_archivo(archivo,
   cas.coef[j]);

  verifica(fprintf(archivo, "%s_{%d}",
   nom_var, j+1));
};
    }
  else /* cas.coef[j]<0 */
    {
      verifica(fprintf(archivo, "-"));
      escribe_fraccionvl_archivo(archivo,
       menosfrvl(cas.coef[j]));
```

```
      verifica(fprintf(archivo,"%s_{%d}", nom_var,
       j+1));
      salio_primer_elemento=1;
    }
}
      else /* cas.coef[j].den=1, esto es, es entero */
{
  if (!es_igualvl(cas.coef[j].num,longint2vl(0))&&
      !cas.coef[j].num.sgn)
    {
      if (!salio_primer_elemento)
{
  if (!es_igualvl(cas.coef[j].num,longint2vl(1)))
    {
      escribevl_archivo(archivo, cas.coef[j].num);
      verifica(fprintf(archivo, "%s_{%d}",
       nom_var, j+1));
      salio_primer_elemento=1;
    }
  else /* cas.coef[j].num=1 */
    {
      verifica(fprintf(archivo, "%s_{%d}",
       nom_var, j+1));
      salio_primer_elemento=1;
    };
}
      else /* salio_primer_elemento */
{
  if (!es_igualvl(cas.coef[j].num,longint2vl(1)))
    {
      verifica(fprintf(archivo, "+"));
      escribevl_archivo(archivo, cas.coef[j].num);
      verifica(fprintf(archivo, "%s_{%d}",
       nom_var, j+1));
    }
  else /* cas.coef[j].num==1 */
    {
      verifica(fprintf(archivo, "+%s_{%d}",
       nom_var, j+1));
      salio_primer_elemento=1;
    }
}
    }
  else /* cas.coef[j].num<0 */
    {
      if (!es_igualvl(cas.coef[j].num,
      menosvl(longint2vl(1))))
{
  escribevl_archivo(archivo, cas.coef[j].num);
  verifica(fprintf(archivo, "%s_{%d}",
    nom_var,
   j+1));
}
      else /* cas.coef[j].num==-1 */
{
  verifica(fprintf(archivo, "-%s_{%d}",
   nom_var, j+1));
```

```
  salio_primer_elemento=1;
};
    };
  salio_primer_elemento=1;
};
    };
};
    };
  verifica(fprintf(archivo, " "));
}

/*void escribe_lista_archivo_maple(triang tr, int fila, FILE *archivo)
{
  int i,j, k;
  lista_de_casillavls lista_de_ceros;
  int num_de_ceros;

  num_de_ceros=0;
  verifica(fprintf(archivo, "# %ld\n", CONTADOR));
  for (i=0;i<=fila-1;i++)
    {
      for (j=0; j<=(i/2); j++)
{
  if (es_cero(tr[i][j]))
    {
      lista_de_ceros[num_de_ceros][0]=i;
      lista_de_ceros[num_de_ceros][1]=j;
      num_de_ceros++;
    }
};
    };
  verifica(fprintf(archivo, "liston:=["));
  for (k=0;k<num_de_ceros;k++)
    {
      if (k>0)
{
  verifica(fprintf(archivo, ", "));
};
*/      /* Notemos que no son iguales los datos exigidos para la salida */
      /* Maple que para la salida de aqui. */
/*      verifica(fprintf(archivo, "[%d,%d]", lista_de_ceros[k][1]+1,
       lista_de_ceros[k][0]+2-lista_de_ceros[k][1]));
    };

  verifica(fprintf(archivo,"];\n\n"));
}
*/

/*void escribe_salida(triang tr, int fila, FILE *archivo)
{
  escribe_triangulo_archivo(tr, fila, archivo);
}
*/

/*void anyadir_a_lista_no_ceros(lista_de_casillavls milista, int numero,
      int f, int c)
{
```

```
  milista[numero][0]=f;
  milista[numero][1]=c;
}

*/
```

A.1.19. El archivo `primosvl.c`

```
/* primovls.c */
/* Funciones sobre primovls que aparecen en la descomposicion de numeros */

#include <stdio.h>
#include "bullet.h"
#include "bulletvl.h"

int es_primovl(verylong n)
{
  return (es_igualvl(n,mayor_primovl(n, MAXPR)));
}

verylong mayor_primovl(verylong  n, verylong maxpr)
{
  verylong m,p, mp;

  if (n.sgn)
    {
      return mayor_primovl(menosvl(n), maxpr);
    }
  else
    {
      if (es_igualvl(n,longint2vl(0)))
{
  return longint2vl(0);
}
      else
{
  mp=longint2vl(1);
  m=n;
  p=longint2vl(2);
  while (es_cerovl(modvl(m,p)))
    {
      m=divvl(m,p);
      mp=p;
    };
  p=longint2vl(1L);
```

```
  while (es_mayorvl(m,longint2vl(1)) &&
 !es_mayorvl(prodvl(p,p),m) && !es_mayorvl(p,maxpr))
    {
      p=sumavl(p,longint2vl(2));
      while (es_cerovl(modvl(m,p)))
{
  m=divvl(m,p);
  mp=p;
};
    };
  if (es_mayorvl(prodvl(p,p),m) && es_mayorvl(m,longint2vl(1)))
    {
      mp=m;
    };
  if (es_mayorvl(p, maxpr))
    {
      if (es_mayorvl(p, MAXPR))
{
  fprintf(stderr, "Primos mayores que MAXPR "
  "detectados...\n");
};
      return maxpr;
    }
  else
    {
      return mp;
    };
};
    };
}

verylong mayor_primovl_casilla(casillavl cas)
{
  int i;
  verylong  m,p;

  if (es_cero_casillavl(cas))
    {
      return(longint2vl(1));
    }
  else
    {
      p=longint2vl(1);
      for (i=0;i<cas.longitud;i++)
{
  m=mayor_primovl(cas.coef[i].num, MAXPR);
  if (es_mayorvl(m,p))
    {
      p=m;
    };
};
      return(p);
    }
}
```

A.1.20. El archivo `despejevl.c`

```
/* Archivo despeje.c, incluye funciones relacionadas con la */
/* substitucion de unas casillavls en otras dentro de los triangulos */
/* del tipo T_G. */

#include <stdio.h>
#include <stdlib.h>
#include "bullet.h"
#include "bulletvl.h"

extern int STACK_OVERFLOW;
casillavl substituirvl(casillavl cs, despejevl desp)
{
  casillavl temp, res, simp, resp;
/*  int i, ind;*/
/*  verylong mcd, mcm;*/
  fraccionvl frtemp/*, fr*/;
  if (cs.longitud!=desp.cas.longitud)
    {
      printf("casillavl y despeje de diferente longitud");
      exit(1);
    };
  if (es_cero_casillavl(cs))
    {
      res=cs;
    }
  else
    {
      /*  printf("CALCULANDO: cs.coef[desp.pos]=(%d/%d), "
  "desp.cas.coef[desp.pos]=(%d/%d), frtemp=(%d/%d)\n",
  cs.coef[desp.pos].num, cs.coef[desp.pos].den,
  desp.cas.coef[desp.pos].num, desp.cas.coef[desp.pos].den,
  frtemp.num, frtemp.den);
  */
/*      printf("Substituyendo\n");
      escribe_casillavl(cs);
      printf("por\n");
      escribe_casillavl(desp.cas);
      printf("a traves de la posicion %d.\n", desp.pos);*/
/*      ind=0;
      while (es_igualvl(cs.coef[ind].num,longint2vl(0)))
{
  ind++;
};
      mcd=cs.coef[ind].num;
      mcm=cs.coef[ind].den;
      for (i=ind+1;i<cs.longitud;i++)
{
  if (!es_cerovl(cs.coef[i].num))
    {
      mcd=gcdvl(cs.coef[i].num,mcd);
      mcm=prodvl((divvl(cs.coef[i].den,
gcdvl(mcm,cs.coef[i].den))), mcm);
    };
};
      fr.num=mcm;
```

```
      fr.den=mcd;
      simp=prodescvl(fr,cs);
*/
      simp=cs;
      frtemp=prodfrvl(simp.coef[desp.pos],
  menosfrvl(inversafrvl(desp.cas.coef[desp.pos])));
      temp=prodescvl(frtemp,desp.cas);
      resp=sum_casvl(temp,simp);
      res=/*prodescvl(inversafrvl(fr),resp)*/ resp;
/*      printf("La casilla se convierte en:\n");
      escribe_casillavl(res);
      printf("\n\n");
*/    }

  return(res);
}

/* El criterio que seguimos para despejar una variable es considerar */
/* el numero de variable que tiene _menor_ su mayor_primovl, tanto del */
/* numerador, como del denominador.  En caso de empate con este */
/* criterio, se substituira la variable mas a la  derecha. */

despejevl despejar_una_variablevl(casillavl c)
{
  despejevl res;
  int i,j, i1;
  verylong mp, mpc;

/*  escribe_casillavl(c);*/
  j=0;
  while (es_cerovl(c.coef[j].num))
    {
      j++;
    };
  i1=j;
  for(i=i1;i<c.longitud; i++)
    {
      if (!es_cerovl(c.coef[i].num))
{
  if (es_mayorvl(absvl(c.coef[i1].num), absvl(c.coef[i].num)))
    {
      i1=i;
    };
};
    };

  mp=mayor_primovl(c.coef[j].num, MAXPR);
  res.pos=j;
  res.cas=c;
  j=0;
  while (es_cerovl(c.coef[j].num))
    {
      j++;
    };
  for (i=j+1;i<c.longitud; i++)
    {
      if (!es_cerovl(c.coef[i].num))
```

```
{
  mpc=mayor_primovl(c.coef[i].num,
    (!es_mayorvl(mp,longint2vl(0)) ? mp :
     MAXPR));
  if (es_mayorvl(mp,mpc))
    {
      mp=mpc;
      res.pos=i;
    };
};
    };
  return (res);
}
```

A.1.21. El archivo `jacobi1.c`

```
#include <stdio.h>
#include <stdlib.h>
#include "bullet.h"
#include "bulletvl.h"

#define MAXJAC 100
#define C_0_MAX (1+c0)

/* el tipo gran_jacobi almacena en una matriz el coeficiente de */
/* x_{i+l}x_{j+l} de uno de los Jacobis como su coeficiente M[i][j]. */
/* Cada uno de sus coeficientes es una fracción muy larga. */

typedef fraccionvl **gran_jacobi;

int STACK_OVERFLOW;
int l, c0;
casillavl *equis;

FILE *salida, *salida_TeX, *salida_cab_TeX, *salida_maple;
/* Van aqui algunas declaraciones previas, antes de construir el */
/* jacobi.h */

void aplica_jacobi(gran_jacobi *jac);
int factorizable_l(gran_jacobi);
int factorizable_l_1(gran_jacobi);
casillavl entre_x_l(gran_jacobi);
casillavl entre_x_l_1(gran_jacobi);
int substituir_formas(casillavl);
int es_cero_jacobi(gran_jacobi);
void jacobi(int i, int j, int k, gran_jacobi *res);
casillavl alpha(int i, int j);
despejevl despejar_final_variablevl(casillavl c);
void inicia_equis();
despejevl despejar_una_variablevl_al_final(casillavl c);
```

```
void escribe_casillavl_archivo_TeX_1(casillavl cas, char *nom_var, FILE
 *archivo);
void escribe_jacobi(gran_jacobi jac);
int aplica_tecnica_2(gran_jacobi *jac, int n);

/* En la siguiente funcion se define la version vl de los numeros */
/* combinatorios.*/


/**************************************************************************/
verylong choosevl(unsigned long int x1, unsigned long int x2)
/**************************************************************************/
/* En esta funcion se define la version verylong de los numeros        */
/* combinatorios.                                                      */
/**************************************************************************/
{
  unsigned long int i;
  verylong res, num;

  STACK_OVERFLOW++;
  if (STACK_OVERFLOW > MAX_STACK_OVERFLOW)
    {
      fprintf(stderr,"Desbordamiento de pila en funci\363n choose.\n");
      exit(1);
    };
  if (x2>x1)
    {
      STACK_OVERFLOW--;
      return (longint2vl(0));
    }
  else
    {
      if (2*x2>x1)
{
  STACK_OVERFLOW--;
  return(choosevl(x1,x1-x2));
}
      else
{
  res=longint2vl(1);
  for (i=0;i<x2;i++)
    {
      num=prodvl(res,longint2vl(x1-i));
      res=divvl(num,longint2vl(i+1));
    };
  STACK_OVERFLOW--;
  return(res);
};
    };
}

/* La siguiente funcion calcula $\alpha_{i,j}$ en funci\'on de las */
/* $x_i$. */

/**************************************************************************/
casillavl alpha(int i, int j)
/**************************************************************************/
```

```
/* Esta funcion devuelve $\alpha_{i,j}$ en funcion de los $x_k$.       */
/*************************************************************************/
{
  casillavl res, sumando, res1;
  int k;
  fraccionvl fr;

  if (i>j)
    {
      return(menos_casvl(alpha(j,i)));
    };

  res.longitud=C_0_MAX;
  for (k=0; k<C_0_MAX; k++)
    {
      res.coef[k].num=longint2vl(0);
      res.coef[k].den=longint2vl(1);
    };
  for (k=(i+j-1)/2; k>=l && k>=i; k--)
    {
      fr.num=(k-i)%2 ? menosvl(choosevl(j-1-k, k-i)):
choosevl(j-1-k, k-i);
      fr.den=longint2vl(1);
      sumando=prodescvl(fr, *(equis+(k-l)));
      res1=sum_casvl(res, sumando);
      res=res1;
    };

  return(res);
}




/*************************************************************************/
void aplica_jacobi(gran_jacobi *jac)
/*************************************************************************/
/* Esta funcion trata de aplicar Jacobi hasta llegar una contradiccion */
/* del tipo x_l=0.                                                       */
/*************************************************************************/
{
  int n, a, b, j;
  verylong p;
  int contradiccion, se_ha_despejado, hay_que_repetir;
  casillavl forma;

  contradiccion=0;
  n=5; /* minimo valor del nivel menos uno */
  while (!contradiccion)
    {
      n++; /* incrementamos el nivel */
      se_ha_despejado=1;
      hay_que_repetir=0;
      fprintf(salida_TeX, "\n\nNivel %d\n\n", n);
      for (a=(n/3)-1; a>0 && !contradiccion; a--)
{
```

```
  for (b=(n-a-1)/2; b>a && !contradiccion; b--)
    {
      jacobi(a,b,n-a-b, jac);
      printf("Jacobi (%d, %d, %d):\n", a, b, n-a-b);
      escribe_jacobi(*jac);
      /* Aqui aplicamos una hipotesis muy fuerte, que es que */
      /* el factor $x_l$ o el factor $x_{l+1}$ se obtiene en */
      /* los Jacobi no nulos que vayan apareciendo.*/
      if (!es_cero_jacobi(*jac))
{
  if (factorizable_l(*jac))
    {
      /* Se factoriza x_l */

      forma=entre_x_l(*jac);
      fprintf(salida_TeX, "\nConsideremos "
      "Jacobi: "
      "$f(%d, %d, %d)=x_{%d}\\bigl(", a, b,
      n-a-b, l);
      escribe_casillavl_archivo_TeX_l(forma, "x",
    salida_TeX);
      fprintf(salida_TeX, "\\bigr)$.\n\n");
      contradiccion=substituir_formas(forma);
      /* El primo... */
      if (!contradiccion)
{
  p=menor_mayor_primo_casillavl(forma);
}
      else
{
  p=mayor_primovl(forma.coef[0].num, MAXPR);
};
      fprintf(salida_TeX, "\nLa substituci\\'on"
      " es v\\'alida para $p>");
      escribevl_archivo(salida_TeX, p);
      fprintf(salida_TeX, "$.\n");
      if (!se_ha_despejado)
{
  hay_que_repetir=1;
};
    }
  else
    {
      if (factorizable_l_1(*jac))
{
  /* Entonces se factoriza por x_{l+1} */

  forma=entre_x_l_1(*jac);

  /* Hipotesis adicional: se llega mas alla */
  /* suponiendo que x_{l+1}\ne 0 que si */
  /* x_{l+1}=0. */
  fprintf(salida_TeX, "\nConsideremos"
  " Jacobi:"
  " $f(%d, %d, %d)=x_{%d}\\bigl(", a, b,
  n-a-b, l+1);
  escribe_casillavl_archivo_TeX_l(forma, "x",
```

```
  salida_TeX);
  fprintf(salida_TeX, "\\bigr)$.\n\n");
  contradiccion=substituir_formas(forma);
  /* El primo... */
  if (!contradiccion)
    {
      p=menor_mayor_primo_casillavl(forma);
    }
  else
    {
      p=mayor_primovl(forma.coef[0].num, MAXPR);
    };
  fprintf(salida_TeX, "\nLa substituci\\'on"
  " es v\\'alida para $p>");
  escribevl_archivo(salida_TeX, p);
  fprintf(salida_TeX, "$.\n");
  if (!se_ha_despejado)
    {
      hay_que_repetir=1;
    };
}
      else
{
  se_ha_despejado=0;
}
    };
};
    };
};
      /* Escribimos los valores validos para el algebra de Lie */
      fprintf(salida_TeX, "\\begin{eqnarray*}\n");
      for (j=0; j<C_0_MAX; j++)
{
  fprintf(salida_TeX, "x_{%d}&=&", j+1);
  escribe_casillavl_archivo_TeX_1(*(equis+j), "x",
  salida_TeX);
  if (j<C_0_MAX-1)
    {
      fprintf(salida_TeX, "\\\\\n");
    };
};
      fprintf(salida_TeX, "\n\\end{eqnarray*}\n\n");
      fflush(salida_TeX);
      if (!se_ha_despejado)
{
  contradiccion=aplica_tecnica_2(jac, n);
  hay_que_repetir=1;
};
      if (hay_que_repetir)
{
  n--;
};
    };
  fprintf(salida_TeX, "\n\nNotemos que, en virtud de la"
  " periodicidad mod $p-1$, que $p>%d$.\n\n", 2*(c0+1)+3);
}
```

```
/**************************************************************************/
void inicia_equis()
/**************************************************************************/
/* En esta funcion se inicializan los valores de las $x_i$ con objeto  */
/* de que puedan ser substituidas por otros valores a lo largo del     */
/* algoritmo.                                                          */
/**************************************************************************/
{
  int i, j;

  vermem(equis=malloc(C_0_MAX*sizeof(casillavl)));
  for (i=0; i<C_0_MAX; i++)
    {
      (equis+i)->longitud=C_0_MAX;
      for (j=0; j<C_0_MAX; j++)
{
  (equis+i)->coef[j].num=longint2vl(i==j);
  (equis+i)->coef[j].den=longint2vl(1);
};
    };
}

/**************************************************************************/
int main(int argc, char *argv[])
/**************************************************************************/
/* Funcion principal del programa.                                        */
/**************************************************************************/
{
/*  verylong p;
  casillavl cas;*/
  gran_jacobi *jac;
  int kkk;
  char *nombre_salida_TeX;

  if (argc<3)
    {
      l=4;
      c0=6;
      printf("Asignados los valores por defecto c0=6, l=4.\n");
    };
  l=atoi(argv[2]);
  c0=atoi(argv[1]);
/*  if (l<=((1+c0)/2))
    {
      fprintf(stderr, "l ha de ser mayor que (c0+1)/2\n");
      exit(1);
    };
*/
  printf("Asignados los valores $c_0=%d$, $l=%d$.\n", c0, l);
  inicia_equis();
  vermem(jac=malloc(sizeof(gran_jacobi)));
  vermem((*jac)=malloc(C_0_MAX*sizeof(fraccionvl*)));
  vermem(nombre_salida_TeX=malloc(12+(l>9?1:0)+(c0>9?1:0)));
  for (kkk=0; kkk<C_0_MAX; kkk++)
    {
      vermem(*(*jac+kkk)=
     malloc((kkk+1)*sizeof(fraccionvl)));
```

```
    };
  sprintf(nombre_salida_TeX, "c_0=%d-l=%d.tex", c0, l);

  if ((salida_TeX=fopen(nombre_salida_TeX, "wt"))==NULL)
    {
      perror("Error en archivo de salida TeX");
      exit(1);
    };
  fprintf(salida_TeX, "Asignados los valores $l=%d$, $c_0=%d$.\n", l,
  c0);
  aplica_jacobi(jac);
  fclose(salida_TeX);
  return(0);
}

/*************************************************************************/
verylong menor_mayor_primo_casillavl(casillavl cas)
/*************************************************************************/
/* Esta funcion devuelve el menor maximo primo presente en una         */
/* casillavl.                                                            */
/*************************************************************************/
{
  despejevl des;

  des=despejar_final_variablevl(cas);
  return(mayor_primovl(des.cas.coef[des.pos].num, MAXPR));
}


/*************************************************************************/
int factorizable_l(gran_jacobi jac)
/*************************************************************************/
/* Devuelve $1$ si $x_l$ es un factor comun de la expresion de Jacobi. */
/* Devuelve $0$ en otro caso.                                            */
/*************************************************************************/
{
  int res, i, j;

  res=1;
  for (i=0; i<C_0_MAX && res; i++)
    {
      for (j=1; j<=i && res; j++)
{
  res=es_cerovl((*(*(jac+i)+j)).num);
};
    };
  return(res);
}

/*************************************************************************/
int factorizable_l_1(gran_jacobi jac)
/*************************************************************************/
/* Devuelve $1$ si $x_{l+1}$ es un factor comun de la expresion de     */
/* Jacobi.  En otro caso, devuelve $0$.                                  */
/*************************************************************************/
{
  int res, i, j;
```

```
  res=1;
  for (i=0; i<C_0_MAX && res; i++)
    {
      for (j=0; j<=i && res; j++)
{
  if (j!=1)
    {
      if (i!=1)
{
  res=es_cerovl((*(*(jac+i)+j)).num);
};
    };
};
    };
  return(res);
}

/*************************************************************************/
casillavl entre_x_l(gran_jacobi jac)
/*************************************************************************/
/* Divide la forma cuadratica del argumento entre $x_{l}$.               */
/*************************************************************************/
{
  casillavl res;
  int i;

  res.longitud=C_0_MAX;
  for (i=0; i<C_0_MAX; i++)
    {
      res.coef[i]=*(*(jac+i));
    };
  return(res);
}


/*************************************************************************/
casillavl entre_x_l_1(gran_jacobi jac)
/*************************************************************************/
/* Divide la forma cuadratica del argumento entre $x_{l+1}$.             */
/*************************************************************************/
{
  casillavl res;
  int i;

  res.longitud=C_0_MAX;
  for (i=1; i<C_0_MAX; i++)
    {
      res.coef[i]=*(*(jac+i)+1);
    };
  res.coef[0]=*(*(jac+1));
  return(res);
}

/*************************************************************************/
int substituir_formas(casillavl cas)
/*************************************************************************/
```

```
/* Despeja una variable de la forma lineal de su argumento y la        */
/* substituye en el vector de las $x_k$  hasta que se anula la         */
/* variable $x_l$, caso en que devuelve $1$, o se substituyen todas,   */
/* caso en que devuelve $0$.                                           */
/***********************************************************************/
{
  int i;
  despejevl des;
  int escero;
  casillavl temp;

  escero=1;
  for (i=1; i<C_O_MAX && escero; i++)
    {
      escero=es_cerovl(cas.coef[i].num);
    };
  if (escero)
    {
      return (1);
    };
  des=despejar_una_variablevl_al_final(cas);

  temp=substituirvl(*equis, des);
  fprintf(salida_TeX, "\n\nRealizamos una substituci\\'on,"
  " a partir de $x_{%d}$,\n$$", des.pos+1);
  escribe_casillavl_archivo_TeX_l(des.cas, "x", salida_TeX);
  fprintf(salida_TeX, "=0$$\n\n");
  *equis=temp;
  if (es_cero_casillavl(*equis))
    {
      return(1);
    }
  else
    {
      for (i=1; i<C_O_MAX; i++)
{
  temp=substituirvl(*(equis+i), des);
  *(equis+i)=temp;
};
      return(0);
    };
}

/***********************************************************************/
int es_cero_jacobi(gran_jacobi jac)
/***********************************************************************/
/* Devuelve $1$ si su argumento es una forma cuadratica nula, o $0$ en */
/* caso contrario.                                                     */
/***********************************************************************/
{
  int res, i, j;

  res=1;
  for (i=0; i<C_O_MAX && res; i++)
    {
      for (j=0; j<=i && res; j++)
{
```

```
  res=es_cerovl((*(*(jac+i)+j)).num);
};
    };
  return(res);
}


/*************************************************************************/
void jacobi(int i, int j, int k, gran_jacobi *res)
/*************************************************************************/
/* Calcula la forma cuadratica de Jacobi $f(i,j,k)$.                     */
/*************************************************************************/
{
  int i1, j1;
  casillavl aij, bij, ajk, bjk, aki, bki;
  fraccionvl f1, f2, f3, f4;

  aij=alpha(i,j);
  bij=alpha(i+j+c0,k);
  ajk=alpha(j,k);
  bjk=alpha(j+k+c0, i);
  aki=alpha(k,i);
  bki=alpha(k+i+c0, j);

  for (i1=0; i1<C_0_MAX; i1++)
    {
      for (j1=0; j1<i1; j1++)
{
  f1=sumfrvl(prodfrvl(aij.coef[i1], bij.coef[j1]),
     prodfrvl(bij.coef[i1], aij.coef[j1]));
  f2=sumfrvl(prodfrvl(ajk.coef[i1], bjk.coef[j1]),
     prodfrvl(bjk.coef[i1], ajk.coef[j1]));
  f3=sumfrvl(prodfrvl(aki.coef[i1], bki.coef[j1]),
     prodfrvl(bki.coef[i1], aki.coef[j1]));
  f4=sumfrvl(f1,f2);
  *(*((*res)+i1)+j1)=sumfrvl(f4,f3);
};
      f1=prodfrvl(aij.coef[i1], bij.coef[i1]);
      f2=prodfrvl(ajk.coef[i1], bjk.coef[i1]);
      f3=prodfrvl(aki.coef[i1], bki.coef[i1]);
      f4=sumfrvl(f1,f2);
      *(*((*res)+i1)+i1)=sumfrvl(f4,f3);

    };
}

/*************************************************************************/
despejevl despejar_final_variablevl(casillavl c)
/*************************************************************************/
/* Despeja una variable de la forma lineal c.  El criterio seguido es  */
/* el de considerar la variable con el menor mayor primo lo mas a la   */
/* derecha posible.                                                      */
/*************************************************************************/
{
  despejevl res;
  int i,j, i1;
  verylong mp, mpc;
```

```
/*  escribe_casillavl(c);*/
  j=1;
  while (es_cerovl(c.coef[j].num))
    {
      j++;
    };
  i1=j;
  for(i=i1;i<c.longitud; i++)
    {
      if (!es_cerovl(c.coef[i].num))
{
  if (es_mayorvl(absvl(c.coef[i1].num), absvl(c.coef[i].num)))
    {
      i1=i;
    };
};
    };

  mp=mayor_primovl(c.coef[j].num, MAXPR);
  res.pos=j;
  for (i=0; i<c.longitud; i++)
    {
      res.cas.coef[i].num=prodvl(c.coef[i].num,longint2vl(1));
      res.cas.coef[i].den=prodvl(c.coef[i].den,longint2vl(1));
    };
  res.cas.longitud=c.longitud;
  j=0;
  while (es_cerovl(c.coef[j].num))
    {
      j++;
    };
  for (i=j+1;i<c.longitud; i++)
    {
      if (!es_cerovl(c.coef[i].num))
{
  mpc=mayor_primovl(c.coef[i].num,
    (!es_mayorvl(mp,longint2vl(0)) ? mp :
     MAXPR));
  if (es_mayorvl(mp,mpc))
    {
      mp=mpc;
      res.pos=i;
    };
};
    };
  return (res);
}

/*************************************************************************/
despejevl despejar_una_variablevl_al_final(casillavl c)
/*************************************************************************/
/* Despeja una variable de la forma lineal c.  El criterio seguido es  */
/* el de considerar la variable con el menor mayor primo lo mas a la   */
/* derecha posible.                                                    */
/*************************************************************************/
{
```

```
  despejevl res;
  int i,j, i1;
  verylong mp, mpc;

/*  escribe_casillavl(c);*/
  j=1;
  while (es_cerovl(c.coef[j].num))
    {
      j++;
    };
  i1=j;
  for(i=i1;i<c.longitud; i++)
    {
      if (!es_cerovl(c.coef[i].num))
{
  if (!es_mayorvl(absvl(c.coef[i].num), absvl(c.coef[i1].num)))
    {
      i1=i;
    };
};
    };

  mp=mayor_primovl(c.coef[j].num, MAXPR);
  res.pos=j;
  for (i=0; i<c.longitud; i++)
    {
      res.cas.coef[i].num=prodvl(c.coef[i].num,longint2vl(1));
      res.cas.coef[i].den=prodvl(c.coef[i].den,longint2vl(1));
    };
  res.cas.longitud=c.longitud;
  j=0;
  while (es_cerovl(c.coef[j].num))
    {
      j++;
    };
  for (i=j+1;i<c.longitud; i++)
    {
      if (!es_cerovl(c.coef[i].num))
{
  mpc=mayor_primovl(c.coef[i].num,
    (!es_mayorvl(mp,longint2vl(0)) ? mp :
     MAXPR));
  if (!es_mayorvl(mpc,mp))
    {
      mp=mpc;
      res.pos=i;
    };
};
    };
  return (res);
}

/*************************************************************************/
void escribe_casillavl_archivo_TeX_l(casillavl cas, char *nom_var, FILE
 *archivo)
/*************************************************************************/
/* Escribe en un archivo TeX la casilla cas, teniendo en cuenta que el */
```

```
/* nombre de las variables es *nom_var_i                              */
/*********************************************************************/
{
  int j;
  int salio_primer_elemento;

  if (es_cero_casillavl(cas))
    {
      verifica(fprintf(archivo, "0"));
    }
  else
    {
      salio_primer_elemento=0;
      for (j=0; j<cas.longitud; j++)
{
  if (!es_cerovl(cas.coef[j].num))
    {
      if (!es_igualvl(cas.coef[j].den,longint2vl(1)))
{
  if (!es_igualvl(cas.coef[j].num,longint2vl(0)) &&
      !cas.coef[j].num.sgn)
    {
      if (!salio_primer_elemento)
{
  escribe_fraccionvl_archivo(archivo, cas.coef[j]);
  verifica(fprintf(archivo,"%s_{%d}", nom_var,
   j+1));
  salio_primer_elemento=1;
}
      else /* ya salio primer elemento */
{
  verifica(fprintf(archivo, "+"));
  escribe_fraccionvl_archivo(archivo,
   cas.coef[j]);

  verifica(fprintf(archivo, "%s_{%d}",
   nom_var, j+1));
};
    }
  else /* cas.coef[j]<0 */
    {
      verifica(fprintf(archivo, "-"));
      escribe_fraccionvl_archivo(archivo,
       menosfrvl(cas.coef[j]));

      verifica(fprintf(archivo,"%s_{%d}", nom_var,
       j+1));
      salio_primer_elemento=1;
    }
}
      else /* cas.coef[j].den=1, esto es, es entero */
{
  if (!es_igualvl(cas.coef[j].num,longint2vl(0))&&
      !cas.coef[j].num.sgn)
    {
      if (!salio_primer_elemento)
{
```

```
  if (!es_igualvl(cas.coef[j].num,longint2vl(1)))
    {
      escribevl_archivo(archivo, cas.coef[j].num);
      verifica(fprintf(archivo, "%s_{%d}",
       nom_var, j+l));
      salio_primer_elemento=1;
    }
  else /* cas.coef[j].num=1 */
    {
      verifica(fprintf(archivo, "%s_{%d}",
       nom_var, j+l));
      salio_primer_elemento=1;
    };
}
      else /* salio_primer_elemento */
{
  if (!es_igualvl(cas.coef[j].num,longint2vl(1)))
    {
      verifica(fprintf(archivo, "+"));
      escribevl_archivo(archivo, cas.coef[j].num);
      verifica(fprintf(archivo, "%s_{%d}",
       nom_var, j+l));
    }
  else /* cas.coef[j].num==1 */
    {
      verifica(fprintf(archivo, "+%s_{%d}",
       nom_var, j+l));
      salio_primer_elemento=1;
    }
}
    }
  else /* cas.coef[j].num<0 */
    {
      if (!es_igualvl(cas.coef[j].num,
      menosvl(longint2vl(1))))
{
  escribevl_archivo(archivo, cas.coef[j].num);
  verifica(fprintf(archivo, "%s_{%d}",
    nom_var,
   j+l));
}
      else /* cas.coef[j].num==-1 */
{
  verifica(fprintf(archivo, "-%s_{%d}",
   nom_var, j+l));
  salio_primer_elemento=1;
};
    };
  salio_primer_elemento=1;
};
    };
};
    };
  verifica(fprintf(archivo, " "));
}
```

```
/**************************************************************************/
void escribe_jacobi(gran_jacobi jac)
/**************************************************************************/
/* Escribe en pantalla la forma cuadratica jac.                           */
/**************************************************************************/
{
  int i,j;

  for (i=0; i<C_0_MAX; i++)
    {
      for (j=0; j<=i; j++)
{
  if (!es_cerovl((*(*(jac+i)+j)).num))
    {
      escribe_fraccionvl(*(*(jac+i)+j));
      printf("x_%d x_%d\n", i+1, j+1);
    };
};
    };
}

/**************************************************************************/
int aplica_tecnica_2(gran_jacobi *jac, int n)
/**************************************************************************/
/* Factoriza expresiones de Jacobi de la forma                            */
/* $ax_l^2+bx_lx_{l+1}+cx_{l+1}^2$.                                       */
/**************************************************************************/
{
  int a, b, i, j;
  int contradiccion, hay_forma_1, despejado;
  casillavl forma1, forma2, forma3;
  despejevl desp1;
  verylong p;

  fprintf(salida_TeX, "\n\nNivel %d\n\nT\\'ecnica de factorizaci\\'on 2\n", n);
  hay_forma_1=0;
  despejado=0;
  contradiccion=0;
  for (a=(n/3)-1; a>0 && !contradiccion && !despejado; a--)
    {
      for (b=(n-a-1)/2; b>a && !contradiccion && !despejado; b--)
{
  jacobi(a,b,n-a-b, jac);
  printf("Jacobi (%d, %d, %d):\n", a, b, n-a-b);
  escribe_jacobi(*jac);
  if (!es_cero_jacobi(*jac))
    {
      if (!hay_forma_1)
{
  forma1.longitud=C_0_MAX;
  for (i=0; i<C_0_MAX; i++)
    {
      forma1.coef[i].num=longint2vl(0);
      forma1.coef[i].den=longint2vl(1);
    };
  forma1.coef[0]=***jac;
```

```
  forma1.coef[1]=*(*(*jac+1));
  forma1.coef[2]=*(*(*jac+1)+1);
  hay_forma_1=1;
  desp1.cas=forma1;
  desp1.pos=2;
}
      else
{
  forma2.longitud=C_0_MAX;
  for (i=0; i<C_0_MAX; i++)
    {
      forma2.coef[i].num=longint2vl(0);
      forma2.coef[i].den=longint2vl(1);
    };
  forma2.coef[0]=***jac;
  forma2.coef[1]=*(*(*jac+1));
  forma2.coef[2]=*(*(*jac+1)+1);
  forma3=substituirvl(forma2,desp1);
  if (!es_cero_casillavl(forma3))
    {
      contradiccion=substituir_formas(forma3);
      despejado=1;
      fprintf(salida_TeX, "\nConsideremos "
      "$0=x_{%d}\\bigl(", l);
      escribe_casillavl_archivo_TeX_l(forma3, "x",
    salida_TeX);
      fprintf(salida_TeX, "\\bigr)$.\n\n");
      /* El primo... */
      if (!contradiccion)
{
  p=menor_mayor_primo_casillavl(forma3);
}
      else
{
  p=mayor_primovl(forma3.coef[0].num, MAXPR);
};
      fprintf(salida_TeX, "\nLa substituci\\'on"
      " es v\\'alida para $p>");
      escribevl_archivo(salida_TeX, p);
      fprintf(salida_TeX, "$.\n\n");
           };
};
    };
};
    };
  /* Escribimos los valores validos para el algebra de Lie */
  fprintf(salida_TeX, "\\begin{eqnarray*}\n");
  for (j=0; j<C_0_MAX; j++)
    {
      fprintf(salida_TeX, "x_{%d}&=&", j+1);
      escribe_casillavl_archivo_TeX_l(*(equis+j), "x",
      salida_TeX);
      if (j<C_0_MAX-1)
{
  fprintf(salida_TeX, "\\\\\n");
};
    };
```

```
  fprintf(salida_TeX, "\n\\end{eqnarray*}\n\n");
  fflush(salida_TeX);
  return (contradiccion);
}
```

A.2. Algoritmo de cálculo de cotas

A.2.1. El archivo `factors.c`

```
/* archivo factors.c */

#include <stdio.h>
#include <stdlib.h>
#include "pjacobi.h"


/*********************************************************************/
int factorizar(gran_jacobi_modp a, long int p, casillamodp *b,
       casillamodp *c)
/*********************************************************************/
/* Trata de dar una factorizacion de la expresion de Jacobi a mod p. */
/* Caso de existir, la coloca en los punteros b y c.  Devuelve un    */
/* entero que indica si dicha solucion existe o no existe.           */
/*********************************************************************/
{
  /* Los valores de entrada son a y p, y la salida se escribe en los */
  /* dos factores b  y c*/
  int especial;
  gran_jacobi_modp d;
  int i, j, k, miembro, t, exito /*, escuadrado */ ;
  long int inv;
  solucion una_solucion;

  especial=-1;
  exito=1;
  una_solucion=malloc(sizeof(psolucion));
  vermem(d=malloc(C_0_MAX*sizeof(long int *)));
  for (i=0; i<C_0_MAX; i++)
    {

      *(d+i)=malloc(C_0_MAX*sizeof(long int));
      for (j=0; j<C_0_MAX; j++)
{
  *(*(d+i)+j)=0;
};
    };
  b->longitud=C_0_MAX;
  c->longitud=C_0_MAX;
  for (i=0; i<C_0_MAX; i++)
    {
      b->coef[i]=0;
      c->coef[i]=0;
    }
```

```
  for (i=0; i<C_0_MAX; i++)
    {
      if ((*(*(a+i)+i))%p)
{
  especial=i;
  i=p;
};
    };
  if (especial!=-1)
    {
      inv=invmodp(*(*(a+especial)+especial), p);
      for (i=0; i<C_0_MAX; i++)
{
  for (j=0; j<=i; j++)
    {
      *(*(d+i)+j)=(((inv*(*(*(a+i)+j)))%p)+p)%p;
    };
};
#ifdef DRAFT
  printf("\nd=\n");
  escribe_jacobi_modp(d);
#endif
      b->coef[especial]=1;
      c->coef[especial]=1;
      for (i=0;i<C_0_MAX;i++)
{
  if (i!=especial)
    {
      miembro=0;
      for (j=0; j<=i; j++)
{
  if (*(*(a+i)+j)%p)
    {
      miembro=1;
      j=p;
    };
};
      if (miembro==0)
{
  for (j=i+1;j<C_0_MAX;j++)
    {
      if (*(*(a+j)+i)%p)
{
  miembro=1;
  j=p;
};
    };
};
      if (miembro==1)
{
  if (i>especial)
    {
      solve2(*(*(a+especial)+especial),
     *(*(a+i)+especial), *(*(a+i)+i), p,
     una_solucion);
    }
  else
```

```
    {
      solve2(*(*(a+especial)+especial),
     *(*(a+especial)+i), *(*(a+i)+i), p,
     una_solucion);
    };
  if (!una_solucion->haysolucion)
    {
      for (k=0; k<C_0_MAX;k++)
{
  free(*(d+k));
};
      free(d);
      free(una_solucion);
      return 0;
    };
  b->coef[i]=(p-una_solucion->sol1)%p;
  c->coef[i]=(p-una_solucion->sol2)%p;
#ifdef DRAFT
  printf("[%ld,%ld]",b->coef[i],c->coef[i]);
  printf("\nd=\n");
  escribe_jacobi_modp(d);
#endif
  exito=1;
  for (j=0; j<i; j++)
    {
      t=*(*(d+i)+j) - (b->coef[i] * c->coef[j]+
       c->coef[i] * b->coef[j]);
      t=((t%p)+p)%p;
      if (t)
{
  exito=0;
  j=p;
};
    };
  if (!exito)
    {
      b->coef[i]=(p-una_solucion->sol2)%p;
      c->coef[i]=(p-una_solucion->sol1)%p;
#ifdef DRAFT
      printf("(cambiado)[%ld,%ld]",b->coef[i],c->coef[i]);
      printf("\nd=\n");
      escribe_jacobi_modp(d);
#endif
      for (j=0;j<i;j++)
{
  t=*(*(d+i)+j)-(b->coef[i] * c->coef[j] +
 c->coef[i] * b->coef[j]);
  t%=p;
  if (t)
    {
#ifdef DRAFT
      printf("\n\n");
#endif
      for (k=0; k<C_0_MAX;k++)
{
  free(*(d+k));
};
```

```
      free(d);
      free(una_solucion);
      return 0;
    };
};
    };
};
    };
}
    }
  else
    {
      for (i=0; i<C_0_MAX; i++)
{
  for (j=0; j<i; j++)
    {
      if ((*(*(a+i)+j))%p)
{
  inv=invmodp(*(*(a+i)+j),p);
  for (k=j;k<=i;k++)
    {
      b->coef[k]=*(*(a+k)+j)*inv;
      b->coef[k]=(((b->coef[k])%p)+p)%p;
    };
  for (k=0; k<j; k++)
    {
      c->coef[k]=*(*(a+i)+k)%p;
      b->coef[k]=(*(*(a+j)+k)*inv)%p;
    };
  j=p;
  i=p;
};
    };
};
      for (i=0; i<C_0_MAX; i++)
{
  for (j=0; j<i; j++)
    {
      t=*(*(a+i)+j)-(b->coef[i] * c->coef[j] + c->coef[i]
     * b->coef[j]);
      t=((t%p)+p)%p;
      if (!t)
{
  for (k=0; k<C_0_MAX;k++)
    {
      free(*(d+k));
    };
  free(d);
  free(una_solucion);
  return 0;
};
    };

};
    };
  for (k=0; k<C_0_MAX;k++)
    {
```

```
      free(*(d+k));
    };
  free(d);

  free(una_solucion);
  return 1;
}
```

A.2.2. El archivo `pjacobi.c`

```
/*********************************************************************/
/* Archivo pjacobi.c, con la funcion main() del programa pjacobi.    */
/*********************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include "pjacobi.h"

void prueba();
gran_jacobi_modp jaco;
/*********************************************************************/
int main(int argc, char*argv[])
/*********************************************************************/
/* Funcion main() que no hace mas que llamar a main_function().      */
/*********************************************************************/
{
  /*  prueba(argc,argv);*/
  /* return 0;*/
  return main_function(argc,argv);
}
```

A.2.3. El archivo `power.c`

```
/*********************************************************************/
/*********************************************************************/
/** Archivo power.c                                                 **/
/** Contiene la definicion de una funcion para calcular potencias   **/
/** modulo p.                                                       **/
/*********************************************************************/
/*********************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include "pjacobi.h"


/*********************************************************************/
long int power(long int base, long int exponente, long int modulo)
/*********************************************************************/
/* Esta funcion calcula la potencia base^exponente y reduce el       */
/* resultado al modulo dado.  Ha sido extraida del libro de Peter    */
```

```
/* Giblin, "Primes and Programming".                                  */
/*********************************************************************/
{
  long int d, res, pow,bas;

  res=1;
  pow=exponente;
  bas=base;
  while (pow > 0)
    {
      d=pow-2*(pow/2);
      if (d==1)
{
  res*=bas;
  res=((res%modulo)+modulo)%modulo;
};
      bas=bas*bas;
      bas=((bas%modulo)+modulo)%modulo;
      pow=(pow-d)/2;
    }
  return res;
}

/*
int main()
{
  long int a, n, m, r;


  printf("Escribe los valores de a, n, m, separados por espacios.\n");
  scanf("%ld %ld %ld",&a, &n, &m);
  r=power(a,n,m);
  printf("El valor de (%ld)^(%ld) mod %ld es %ld.\n", a,n,m,r);
  return 0;
}

*/
```

A.2.4. El archivo `legendre.c`

```
/*********************************************************************/
/* Archivo legendre.c                                                */
/* En este archivo se definen funciones para el calculo de simbolos  */
/* de Legendre/Jacobi                                                */
/*********************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include "pjacobi.h"



/*********************************************************************/
void WriteRatio(long int n, long int k, int sign)
/*********************************************************************/
```

```
/* Esta funcion escribe en pantalla de una manera simplificada los   */
/* simbolos de Legendre/Jacobi que aparecen en el calculo.  Solo se  */
/* usa cuando esta definido DRAFTSHANKS.                             */
/*********************************************************************/
{
  if (sign==-1)
    {
      printf("-[ %ld  /  %ld ]\n",n,k);
    }
  else
    {
      printf("+[ %ld  /  %ld ]\n",n,k);
    }
} /*WriteRatio*/



/*********************************************************************/
int legendre(long int ene, long int ka)
/*********************************************************************/
/* Esta funcion devuelve el simbolo de Legendre/Jacobi ${n\brack     */
/* k}$. La idea del algoritmo es del texto de Giblin.                */
/*********************************************************************/
{
  long int n,k,count,x, y,  xtempk;
  int sign;


  n=ene;
  k=ka;

  sign=1;
  while (n>1)
    {
      n%=k;
#ifdef DRAFTSHANKS
      WriteRatio(n,k,sign);
#endif
      count=0;

      while (!(n%2)) /* o sea, cuando n es par */
{
  n/=2;
  count++;
};
      if (count%2) /* o sea, count es impar */
{
  x=k%8;
  if ((x==3)||(x==5))
    {
      sign*=-1; /* propiedades del simbolo $2\brack p$ */
    };
};
#ifdef DRAFTSHANKS
      if (count > 0)
{
  WriteRatio(n,k,sign);
```

```
};
#endif
      if (n>1)
{
  xtempk=k;
  k=n;
  n=xtempk;
  /* Hemos intercambiado n y k */
  x=k%4;
  y=n%4;
  if (x==3)
    {
      if (y==3)
{
  sign*=-1;
}
    };
#ifdef DRAFTSHANKS
  WriteRatio(n,k,sign);
#endif
};
    }
  return sign;
}
```

A.2.5. El archivo shanks.c

```
/* Archivo shanks.c */
/* Contiene el algoritmo de Shanks en la version descrita en el texto */
/* de Giblin para calcular las raices cuadradas en el cuerpo de $p$ */
/* elementos.*/


#include <stdio.h>
#include <stdlib.h>
#include "pjacobi.h"

/* La funcion shanks_sqrt_intermedio determina la raiz cuadrada de a */
/* dado el valor inicial z del algoritmo de Shanks.*/

/*********************************************************************/
long int shanks_sqrt_intermedio(long int a,long int p,long int z)
/*********************************************************************/
/* La funcion shanks_sqrt_intermedio determina la raiz cuadrada de a */
/* dado el valor inicial z del algoritmo de Shanks.                  */
/*********************************************************************/
{
  long int b, c, p1, n, odd, k1, x, y, looplength, i, index, s;


  p1=p-1;
  index=0;
  while(!(p1%2)) /* p1 par */
    {
```

```
      p1/=2;
      index++;
    };
  s=index;
  odd=p1; /* El numero 2k+1 del texto */
  c=power(z,odd,p);
  n=power(a,odd,p);
#ifdef DRAFTSHANKS
  printf("n=%ld\n",n);
#endif
  k1=(odd+1)/2; /* k+1 */
  x=power(a,k1,p);
#ifdef DRAFTSHANKS
  printf("x=%ld\n",x);
#endif
  while (n>1)
    {
      looplength=s;
      y=n; /* y sera la variable que se eleva al cuadrado */
      for (i=1;i<=looplength;i++)
{
  if (y==1)
    {
      y=c;
      s=i-1;
    }
  else
    {
      y=y*y;
      y=((y%p)+p)%p;
    };
};
      b=y;
      c=b*b;
      c=((c%p)+p)%p;
      x=b*x;
      x=((x%p)+p)%p;
#ifdef DRAFTSHANKS
      printf("x=%ld\n", x);
#endif
      n*=c;
      n=((n%p)+p)%p;
#ifdef DRAFTSHANKS
      printf("n=%ld\n", n);
#endif
    };
  return x;
}


/*********************************************************************/
long int shanks_inicial(long int p)
/*********************************************************************/
/* Obtiene un valor inicial para el algoritmo de Shanks, esto es, un */
/* no residuo cuadratico.                                            */
/*********************************************************************/
{
```

```
  long int z;

  for (z=2;;z++)
    {
      if (legendre(z,p)==-1) break;
    };
  return z;
}

/*********************************************************************/
long int shanks_sqrt(long int a, long int p)
/*********************************************************************/
/* Algoritmo de Shanks para el calculo de la raiz cuadrada mod p     */
/*********************************************************************/
{
  long int z, res;
  if (a%p==0)
    {
      return 0;
    }
  else
    {
      z=shanks_inicial(p);
      res=shanks_sqrt_intermedio(a,p,z);
      return res;
    };

}
```

A.2.6. El archivo `modular.c`

```
/* Archivo modular.c, que tiene definidas funciones para trabajar mod */
/* p. Las funciones para invertir mod p han sido hurtadas de Peter    */
/* Giblin, "Primes and Programming", Cambridge,  pag. 21.             */

#include <stdio.h>
#include <stdlib.h>
#include "pjacobi.h"

/*********************************************************************/
long int choose_modp(unsigned long int x1, unsigned long int x2)
/*********************************************************************/
/* Esta funcion calcula el numero combinatorio ${x1\choose x2}$ mod p. */
/*********************************************************************/
{
  unsigned long int i;
  long int res, num;

  STACK_OVERFLOW++;
  if (STACK_OVERFLOW > MAX_STACK_OVERFLOW)
    {
      fprintf(stderr,"Desbordamiento de pila en funci\363n choose_modp.\n");
      exit(1);
    };
  if (x2>x1)
```

```
    {
      STACK_OVERFLOW--;
      return ((unsigned long) 0);
    }
  else
    {
      if (2*x2>x1)
{
  res=choose_modp(x1,x1-x2);
  STACK_OVERFLOW--;
  return res;
}
      else
{
  res=(unsigned long int) 1;
  for (i=0;i<x2;i++)
    {
      num=(((res*(x1-i)) % primo)+primo)%primo;
      res=(((num*invmodp(i+1, primo)) % primo)+primo)%primo;
    };
  STACK_OVERFLOW--;
  return(res);
};
    };
}


/*********************************************************************/
EUCLIDES eucgcd(long int a0, long int b0)
/*********************************************************************/
/* Esta funcion calcula el maximo comun divisor de dos numeros       */
/* enteros, a0 y b0, asi como los coeficientes que aparecen en la    */
/* relacion de Bezout.  Estos tres datos aparecen reflejados en el   */
/* tipo EUCLIDES.  El algoritmo empleado es el de Euclides, en la    */
/* version descrita en lenguaje PASCAL por Peter Gibling, "Primes and  */
/* Programming", Cambridge, en la pagina 21.                         */
/*********************************************************************/
{
  EUCLIDES res;
  long int a, b, q, r, s, s1, s2, t, t1, t2;

  a=a0;
  b=b0;
  s2=1;
  s1=0;
  t2=0;
  t1=1;
  while (b>0)
    {
      q=a/b;
      r=a-b*q;
      s=s2-q*s1;
      t=t2-q*t1;
      /* Tenemos que a0*s+b0*t=r */
      a=b;
      b=r;
      s2=s1;
```

```
      s1=s;
      t2=t1;
      t1=t;
      /* Aqui, el nuevo s2 es igual al viejo s1, y asi sucesivamente, */
      /* antes de pasar al siguiente paso */
    }
/*  printf("El mcd de %ld y %ld es %ld.\n", a0, b0, a);*/
/*  printf("%ld*%ld+%ld*%ld=%ld\n", a0, s2, b0, t2, a);*/


  res.g=a;
  res.lambda=s2;
  res.mu=t2;
  return(res);

}

/*************************************************************************/
long int invmodp(long int a0, long int p)
/*************************************************************************/
/* En esta funcion se obtiene el inverso de a0 mod p.  Hace uso de la  */
/* funcion definida anteriormente, y, en el caso de que p no sea primo */
/* y $\gcd(a0, p)\ne 1$, se genera un mensaje de error.                */
/*************************************************************************/
{
  EUCLIDES euc;
  euc=eucgcd(a0, p);
  if (euc.g!=1)
    {
      fprintf(stderr, "\007%ld no es invertible mod %ld.\n", a0, p);
    };
  return ((euc.lambda+p)%p);
}




/*************************************************************************/
casillamodp substituir_modp(casillamodp cs, despejemodp desp)
/*************************************************************************/
/* Substituye en una casillamodp una variable despejada mod p de otra  */
/* casilla.                                                            */
/*************************************************************************/
{
  casillamodp res, resp;
  int i;
  long int simp;

  if (cs.longitud!=desp.cas.longitud)
    {
      printf("casillamodp y despejemodp de diferente longitud");
      exit(1);
    };
  if (es_cero_cas_modp(cs))
    {
      for (i=0; i<cs.longitud; i++)
{
```

```
      res.coef[i]=cs.coef[i];
}
      res.longitud=cs.longitud;
    }
  else
    {
      simp=((primo-1) * invmodp(desp.cas.coef[desp.pos], primo)
    * (cs.coef[desp.pos])) %primo;
      resp=prodesc_modp(simp, desp.cas);
      res=sum_cas_modp(cs, resp);

    }

  return(res);
}


/*************************************************************************/
despejemodp despejar_una_variablemodp(casillamodp c)
/*************************************************************************/
/* En esta funcion se intenta despejar una variable (mod p) de una     */
/* casilla.  Despejamos la variable situada mas a la derecha.          */
/*************************************************************************/
{
  despejemodp res;
  int i,j;

  res.pos=0;
  for (i=0; i<c.longitud; i++)
    {
      res.cas.coef[i]=c.coef[i];
    };
  res.cas.longitud=c.longitud;
  j=0;
  while ((c.coef[j]%primo)==0)
    {
      j++;
    };
  res.pos=j;
  for (i=j+1;i<c.longitud; i++)
    {
      if ((c.coef[i]%primo)!=0)
{
  res.pos=i;

};
    };
  return (res);
}




/*************************************************************************/
casillamodp prodesc_modp(long int escalar, casillamodp cas)
/*************************************************************************/
```

```
/* En esta funcion se da el resultado de multiplicar un escalar (un    */
/* entero mod p) por una casilla (esto es, una forma lineal).          */
/***********************************************************************/
{
  unsigned int iii;
  casillamodp res;
  long int el, nuevo_el;

  res.longitud=cas.longitud;
  for (iii=0; iii<cas.longitud; iii++)
    {
      el=cas.coef[iii];
      nuevo_el=( ( (escalar*el) %primo) +primo) %primo;
      res.coef[iii]=nuevo_el;
    };
  return(res);
}

/***********************************************************************/
casillamodp menos_cas_modp(casillamodp cas)
/***********************************************************************/
/* Multiplica por -1 la casillamodp cas.                               */
/***********************************************************************/
{
  return(prodesc_modp(primo-1,cas));
}

/***********************************************************************/
casillamodp sum_cas_modp(casillamodp cas1, casillamodp cas2)
/***********************************************************************/
/* Suma dos casillas mod p.                                            */
/***********************************************************************/
{
  unsigned char i;
  casillamodp res;
  if (cas1.longitud!=cas2.longitud)
    {
      fprintf(stderr, "Error, solo esta definida la suma de"
      " \"casillamodps\" de igual longitud");
      exit(1);
    }
  else
    {
      res.longitud=cas1.longitud;
      for (i=0;i<cas1.longitud;i++)
{
  res.coef[i]=(cas1.coef[i] + cas2.coef[i]) % primo;
}
      return (res);
    };
}


/***********************************************************************/
int es_cero_cas_modp(casillamodp cas)
/***********************************************************************/
/* Devuelve 1 si la casilla es nula, 0 en caso contrario.              */
```

```
/*********************************************************************/
{
  int escero;
  int i;
  escero=1;
  for (i=0;
       i<cas.longitud;
       i++)
    {
      if (escero)
{
  escero=((cas.coef[i]%primo)==0);
};
    };
  return(escero);
}



/*********************************************************************/
void escribe_casillamodp(casillamodp c)
/*********************************************************************/
/* Escribe en stdout una casilla mod p.                              */
/*********************************************************************/
{
  int i;
  for (i=0;i<c.longitud;i++)
    {
      printf("%ld ",(c.coef[i]));
    };
}

/*********************************************************************/
int son_casillamodps_iguales(casillamodp cas1, casillamodp cas2)
/*********************************************************************/
/* Devuelve $1$ si las dos casillas son iguales, $0$ en otro caso.   */
/*********************************************************************/
{
  int i, res;
  if (cas1.longitud!=cas2.longitud)
    {
      return 0;
    }
  else
    {
      res=1;
      for (i=0; (i<cas1.longitud) && res; i++)
{
  res= (cas1.coef[i]==cas2.coef[i]);
};
      return(res);
    };
}

/*********************************************************************/
casillamodp alpha_modp(int i, int j)
/*********************************************************************/
```

```
/* Calcula $\alpha_{i,j}$ mod p en funcion de los $x_k$.              */
/*********************************************************************/
{
  casillamodp res, sumando, res1;
  int k;
  long int fr;

/*
  if ((i>=primo)||(j>=primo))
    {
      return(alpha_modp(((i-1)%(primo-1))+1,((j-1)%(primo-1))+1));
    };
*/
  if (i>j)
    {
      return(menos_cas_modp(alpha_modp(j,i)));
    };

  res.longitud=C_0_MAX;
  for (k=0; k<C_0_MAX; k++)
    {
      res.coef[k]=0;
    };
  for (k=(i+j-1)/2; k>=l && k>=i; k--)
    {
      fr=(k-i)%2 ?
(((primo-1)*((choose_modp(j-1-k, k-i)) % primo) ) % primo) :
  (choose_modp(j-1-k, k-i) % primo);
      sumando=prodesc_modp(fr, *(equis+(k-l)));
      res1=sum_cas_modp(res, sumando);
      res=res1;
    };

  return(res);
}

/*********************************************************************/
void inicia_equis_modp(void)
/*********************************************************************/
/* Inicializa el vector de las $x_i$, haciendo cada elemento igual a  */
/* la $x_i$ correspondiente, y aplicando despues, mientras sea        */
/* posible, la periodicidad mod p-1.                                  */
/*********************************************************************/
{
  int i, j, i1, i2/*, a*/;
  casillamodp el_alpha;

  vermem(equis=malloc(C_0_MAX*sizeof(casillamodp)));
  for (i=0; i<C_0_MAX; i++)
    {
      (equis+i)->longitud=C_0_MAX;
      for (j=0; j<C_0_MAX; j++)
{
  (equis+i)->coef[j]=(i==j);
};
    };
  /* El siguiente paso consiste en intentar aplicar la periodicidad */
```

```
  /* modulo p-1 en los casos en que sea posible.  Las variables hasta */
  /* (p-3)/2 son todas basicas, las demas variables aparecen ligadas.*/
  for (i=1;i<=(primo+1)/2; i++)
    {
      i1=i;
      i2=i1+primo-1;
      el_alpha=alpha_modp(i1, i2);
      /* if (i1>=l)*/
      /*{*/
      /*a=(el_alpha.coef[i1-l]+(primo-1)) % primo;*/
      /*  el_alpha.coef[i1-l]=a;*/
      /*  };*/
      /*  */
      substituir_formas_modp(el_alpha);
    }
  /*  el_alpha=alpha_modp(primo-1,1);*/
  /*a=(el_alpha.coef[primo-2]+(primo-1)) % primo;*/
  /*el_alpha.coef[primo-2]=a;*/
  /*substituir_formas_modp(el_alpha);*/
  /* */
  /*  if ((primo-1)/2-l<C_0_MAX)*/
  /*  {*/
  /*    el_alpha=alpha_modp(1, primo);*/
  /*substituir_formas_modp(el_alpha);*/
  /*    };*/
  /* Escribimos los valores validos para el algebra de Lie */
#ifndef SIN_SALIDA
  fprintf(salida_TeX, "\\'Estos son los valores iniciales:\n"
  "\\begin{eqnarray*}\n");
  for (j=0; j<C_0_MAX; j++)
    {
      fprintf(salida_TeX, "x_{%d}&=&", j+l);
      escribe_casillamodp_archivo_TeX_l(*(equis+j), "x",
salida_TeX);
      if (j<C_0_MAX-1)
{
  fprintf(salida_TeX, "\\\\\n");
};
    };
  fprintf(salida_TeX, "\n\\end{eqnarray*}\n\n");
  fflush(salida_TeX);
#endif
}

/*********************************************************************/
int factorizable_l_modp(gran_jacobi_modp jac)
/*********************************************************************/
/* Devuelve $1$ si la relacion de Jacobi tiene como factor comun       */
/* $x_l$, y $0$ en otro caso.                                          */
/*********************************************************************/
{
  int res, i, j;

  res=1;
  for (i=0; i<C_0_MAX && res; i++)
    {
      for (j=1; j<=i && res; j++)
```

```
{
  res=((*(*(jac+i)+j))==0);
};
    };
  return(res);
}



/**************************************************************************/
casillamodp entre_x_1_modp(gran_jacobi_modp jac)
/**************************************************************************/
/* Trata de dividir el Jacobi entre $x_1$, supuesto que $x_1$ es       */
/* factor comun.                                                          */
/**************************************************************************/
{
  casillamodp res;
  int i;

  res.longitud=C_0_MAX;
  for (i=0; i<C_0_MAX; i++)
    {
      res.coef[i]=*(*(jac+i));
    };
  return(res);
}


/**************************************************************************/
int substituir_formas_modp(casillamodp cas)
/**************************************************************************/
/* Dada una casilla, despeja una variable y substituye el resultado en */
/* el vector de las $x_i$ hasta que $x_1=0$.  En este caso, devuelve   */
/* $1$, en otro caso, substituye todas las variables $x_i$ y devuelve  */
/* $0$.                                                                   */
/**************************************************************************/
{
  int i;
  despejemodp des;
  int escero, res;
  casillamodp temp;


  escero=1;
  res=0;
  for (i=0; i<C_0_MAX && escero; i++)
    {
      escero=(0==cas.coef[i]);
    };
  if (escero)
    {
      res=1;
    };
  des=despejar_una_variablemodp(cas);

  temp=substituir_modp(*equis, des);
#ifndef SIN_SALIDA
```

```
  fprintf(salida_TeX, "\n\nRealizamos una substituci\\'on,"
  " a partir de $x_{%d}$,\n$$", des.pos+1);
  escribe_casillamodp_archivo_TeX_1(des.cas, "x", salida_TeX);
  fprintf(salida_TeX, "=0$$\n\n");
#endif
  *equis=temp;
  if (es_cero_cas_modp(*equis))
    {
      res=1;
    }
  else
    {
      for (i=1; i<C_0_MAX; i++)
{
  temp=substituir_modp(*(equis+i), des);
  *(equis+i)=temp;
};
      res=0;
    };
  return res;
}

/*************************************************************************/
int es_cero_jacobi_modp(gran_jacobi_modp jac)
/*************************************************************************/
/* Devuelve $1$ si el Jacobi correspondiente es nulo, $0$ en caso      */
/* contrario.                                                            */
/*************************************************************************/
{
  int res, i, j;

  res=1;
  for (i=0; i<C_0_MAX && res; i++)
    {
      for (j=0; j<=i && res; j++)
{
  res=(0==(*(*(jac+i)+j)));
};
    };
  return(res);
}

/*************************************************************************/
void jacobi_modp( int i, int j, int k, gran_jacobi_modp *res)
/*************************************************************************/
/* Asigna a *res los coeficientes de la forma cuadratica f(i,j,k).     */
/*************************************************************************/
{
  int i1, j1;
  casillamodp aij, bij, ajk, bjk, aki, bki;
  long int f1, f2, f3, f4;

  aij=alpha_modp(i,j);
  bij=alpha_modp(i+j+c0,k);
  ajk=alpha_modp(j,k);
  bjk=alpha_modp(j+k+c0, i);
  aki=alpha_modp(k,i);
```

```
  bki=alpha_modp(k+i+c0, j);

  for (i1=0; i1<C_0_MAX; i1++)
    {
      for (j1=0; j1<i1; j1++)
{
  f1=(((aij.coef[i1] * bij.coef[j1])%primo)
      + ((bij.coef[i1] * aij.coef[j1]) % primo)) % primo;
  f2=(((ajk.coef[i1] * bjk.coef[j1])%primo)
      + ((bjk.coef[i1] * ajk.coef[j1]) % primo)) % primo;
  f3=(((aki.coef[i1] * bki.coef[j1])%primo)
      + ((bki.coef[i1] * aki.coef[j1]) % primo)) % primo;
  f4=(f1+f2) % primo;
  *(*((*res)+i1)+j1)=(f4+f3) % primo;
};
      f1=(aij.coef[i1] * bij.coef[i1]) % primo;
      f2=(ajk.coef[i1] * bjk.coef[i1]) % primo;
      f3=(aki.coef[i1] * bki.coef[i1]) % primo;
      f4=(f1 + f2) %  primo;
      *(*((*res)+i1)+i1)=(f4 + f3) % primo;

    };
}


/**************************************************************************/
void escribe_casillamodp_archivo_TeX_l(casillamodp cas, char *nom_var,
       FILE *archivo)
/**************************************************************************/
/* Escribe en un archivo TeX el valor de una casilla, denotando con el */
/* nombre *nom_var las diversas variables.                              */
/**************************************************************************/
{
  int j;
  int salio_primer_elemento;

  if (es_cero_cas_modp(cas))
    {
      verifica(fprintf(archivo, "0"));
    }
  else
    {
      salio_primer_elemento=0;
      for (j=0; j<cas.longitud; j++)
{
  if ((cas.coef[j]%primo)!=0)
    {
      if (!salio_primer_elemento)
{
  if (((cas.coef[j]-1)%primo)!=0)
    {
      verifica(fprintf(archivo, "%ld" ,cas.coef[j]));
      verifica(fprintf(archivo, "%s_{%d}",
       nom_var, j+1));
      salio_primer_elemento=1;
    }
  else /* cas.coef[j]=1 */
```

```
    {
      verifica(fprintf(archivo, "%s_{%d}",
       nom_var, j+1));
      salio_primer_elemento=1;
    };
}
      else /* salio_primer_elemento */
{
  if (((cas.coef[j]-1)%primo)!=0)
    {
      verifica(fprintf(archivo, "+%ld", cas.coef[j]));
      verifica(fprintf(archivo, "%s_{%d}",
       nom_var, j+1));

    }
  else /* cas.coef[j].num==1 */
    {
      verifica(fprintf(archivo, "+%s_{%d}",
       nom_var, j+1));
    }
}
    }
};
    };
  verifica(fprintf(archivo, " "));
}


/************************************************************************/
void escribe_jacobi_modp(gran_jacobi_modp jac)
/************************************************************************/
/* Escribe en pantalla el valor de un Jacobi.                           */
/************************************************************************/
{
  int i,j;

  for (i=0; i<C_0_MAX; i++)
    {
      for (j=0; j<=i; j++)
{
  if (((*(*(jac+i)+j))%primo)!=0)
    {
      printf("%ld",(*(*(jac+i)+j)));
      printf("x_%d x_%d\n", i+1, j+1);
    };
};
    };
}



/************************************************************************/
void verifica(int valor)
/************************************************************************/
/* Esta funcion detiene el programa si su argumento, una funcion de     */
/* escritura, devuelve un numero negativo.                              */
/************************************************************************/
```

```
{
  if (valor<0)
    {
      perror("Error de escritura");
      exit(1);
    };
}
```

A.2.7. El archivo quadr.c

```
/*********************************************************************/
/* Archivo quad.c                                                    */
/* Contiene las definiciones de funciones necesarias para resolver   */
/* una ecuacion de segundo grado mod p.                              */
/*********************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include "pjacobi.h"

/*********************************************************************/
void solve2(long int a, long int b, long int c, long int p,
    solucion res)
/*********************************************************************/
/* Esta funcion obtiene la solucion de la ecuacion de segundo grado  */
/* $ax^2+bx+c=0$ en el cuerpo $Z_p$ (supuesto que sea resoluble) y   */
/* coloca la solucion en res.  Caso de no ser resoluble, devuelve un */
/* puntero con un indicador de que no hay solucion.                  */
/*********************************************************************/
{
  long int disc, r, inverso, b1,c1;

  if (!((2*a)%p)) /* o sea, 2a es cong con 0 mod p */
    {
      res->haysolucion=0;
      res->sol1=0;
      res->sol2=0;
    }
  else
    {
      disc=b*b-4*a*c;
      disc=((disc%p)+p)%p;
      if (disc)
{
  if (legendre(disc, p)==1)
    {
      /* Hay solucion */
      r=shanks_sqrt(disc,p);
      inverso=invmodp(2*a,p);
      b1=(((inverso*(-b+r))%p)+p)%p;
      c1=(((inverso*(-b-r))%p)+p)%p;
      res->haysolucion=1;
      res->sol1=b1;
```

```
      res->sol2=c1;
    }
  else
    {
      res->haysolucion=0;
      res->sol1=0;
      res->sol2=0;
    };
}
      else
{
  res->haysolucion=1;
  inverso=invmodp(2*a,p);
  res->sol1=(inverso*(p-b))%p;
  res->sol2=res->sol1;
};
    };
}
```

A.2.8. El archivo `jacarbol.c`

```
/* Archivo jacarbol.c */

#include <stdio.h>
#include <stdlib.h>
#include "pjacobi.h"
/* Si se define */
/* #define REQUETEVERBOSE 1*/
/* se ve en la salida estandar como va el calculo. */

#ifdef LINEAL
void procesar_jacobi_no_fact(gran_jacobi_modp jac);
#endif

/*********************************************************************/
int main_function(int argc, char *argv[])
/*********************************************************************/
/* Funcion principal de este programa.  Contiene la descripcion      */
/* general del algoritmo.                                            */
/*********************************************************************/
{
  gran_jacobi_modp *jac;
  int kkk;
  char *nombre_salida_TeX;

  if (argc<4)
    {
      l=4;
      c0=6;
      primo=19;
      printf("Asignados los valores por defecto $c0=6$, $l=4$, $p=19$.\n");
    }
  else
    {
```

```
      l=atoi(argv[2]);
      c0=atoi(argv[1]);
      primo=atol(argv[3]);
      printf("Asignados los valores $c_0=%d$, $l=%d$, $p=%ld$.\n", c0,
     l, primo);
    };
  C_0_MAX=primo-l;
  vermem(jac=malloc(sizeof(gran_jacobi_modp)));
  vermem((*jac)=malloc(C_0_MAX*sizeof(long int *)));
  vermem(nombre_salida_TeX=malloc(22));
  for (kkk=0; kkk<C_0_MAX; kkk++)
    {
      vermem(*(*jac+kkk)=
     malloc((kkk+1)*sizeof(long int)));
    };
  sprintf(nombre_salida_TeX, "c_0=%d-l=%d-p=%ld.tex", c0, l, primo);

  if ((salida_TeX=fopen(nombre_salida_TeX, "wt"))==NULL)
    {
      perror("Error en archivo de salida TeX");
      exit(1);
    };
  fprintf(salida_TeX, "Asignados los valores $l=%d$,"
  " $c_0=%d$, $p=%ld$.\n", l, c0, primo);
  inicia_equis_modp();
  ene=(C_0_MAX)*(C_0_MAX+1)/2;
#ifdef REQUETEVERBOSE
  printf("El valor de ene es %ld.\n", ene);
#endif
  aplica_jacobi_modp(jac);
  fclose(salida_TeX);
  for (kkk=0; kkk<C_0_MAX; kkk++)
    {
      free(*(*jac+kkk));
    };
  free (*jac);
  free(jac);
  return(0);
}



/*************************************************************************/
void aplica_jacobi_modp(gran_jacobi_modp *jac)
/*************************************************************************/
/* Aplica las relaciones de Jacobi para los distintos niveles tratando */
/* de factorizarlas hasta  conseguir una contradiccion.                */
/*************************************************************************/
{
  int  a, b, j, todos_los_jacobis_son_nulos;
  casillamodp forma;

  contradiccion=0;
  nivel_de_reserva=5;
  maximo_nivel_hallado=5;
  hay_que_repetir=0;
  nivel=5; /* minimo valor del nivel menos uno */
```

```
  inicia_superpila();
  for (;;)
    {
      while (!contradiccion)
{
  nivel++; /* incrementamos el nivel */
  hay_jacobi_despejado=0;
  se_ha_despejado=1;
  todos_los_jacobis_son_nulos=1;
#ifdef REQUETEVERBOSE
  printf("\n\nNivel actual: %d\nNivel de reserva:"
 " %d\nMaximo nivel hallado: %d\n", nivel,
 nivel_de_reserva, maximo_nivel_hallado);
#endif
  if (maximo_nivel_hallado<nivel)
    {
      maximo_nivel_hallado=nivel;
    };
  se_ha_despejado=1;
  fprintf(salida_TeX, "\n\nNivel %d\n\n", nivel);
  for (a=(nivel/3)-1; a>0 && !contradiccion; a--)
    {
      for (b=(nivel-a-1)/2; b>a && !contradiccion; b--)
{
  jacobi_modp(a,b,nivel-a-b, jac);
#ifdef REQUETEVERBOSE
  printf("Jacobi (%d, %d, %d):\n", a, b, nivel-a-b);
  escribe_jacobi_modp(*jac);
#endif
  if (!es_cero_jacobi_modp(*jac))
    {
      todos_los_jacobis_son_nulos=0;
      if (factorizable_l_modp(*jac))
{
  /* Se factoriza x_l */
#ifdef REQUETEVERBOSE
  printf("** Factorizacion tipo 1 **\n");
#endif
  forma=entre_x_l_modp(*jac);
  fprintf(salida_TeX, "\nConsideremos "
  "Jacobi: "
  "$f(%d, %d, %d)=x_{%d}\\bigl(", a, b,
  nivel-a-b, l);
  escribe_casillamodp_archivo_TeX_l(forma, "x",
    salida_TeX);
  fprintf(salida_TeX, "\\bigr)$.\n\n");
  contradiccion=substituir_formas_modp(forma);
  hay_jacobi_despejado=1;
  if (!se_ha_despejado)
    {
      hay_que_repetir=1;
    };
}
      else
{
  se_ha_despejado=0;
```

```
};
    };
};
    };
  if ((hay_jacobi_despejado)&&(!se_ha_despejado))
    {
      hay_que_repetir=1;
    };
  /* Escribimos los valores validos para el algebra de Lie */
  fprintf(salida_TeX, "\\begin{eqnarray*}\n");
  for (j=0; j<C_0_MAX; j++)
    {
      fprintf(salida_TeX, "x_{%d}&=&", j+1);
      escribe_casillamodp_archivo_TeX_1(*(equis+j), "x",
salida_TeX);
      if (j<C_0_MAX-1)
{
  fprintf(salida_TeX, "\\\\\n");
};
    };
  fprintf(salida_TeX, "\n\\end{eqnarray*}\n\n");
  fflush(salida_TeX);
  if (!hay_jacobi_despejado&&!todos_los_jacobis_son_nulos)
    {
#ifdef REQUETEVERBOSE
      printf("Imposible despejar de momento.\n");
#endif
      contradiccion=aplica_tecnica_2_modp(jac, nivel);

      hay_que_repetir=1;
    };
  if (hay_jacobi_despejado&&hay_que_repetir)
    {
      nivel=nivel_de_reserva;
      hay_jacobi_despejado=0;
      hay_que_repetir=0;
    };
  if (hay_jacobi_despejado&&!hay_que_repetir)
    {
      nivel_de_reserva=nivel;
    };
  if
    (todos_los_jacobis_son_nulos &&
     (nivel_de_reserva==nivel-1))
      {
nivel_de_reserva++;
      };
};
      if (superpila->siguiente==NULL)
{
  break;
};
      restaura_superpila();
      nivel--;
      contradiccion=0;
      continue;
    };
```

```
  fprintf(salida_TeX, "\nLa pila ha quedado vaciada del todo.\n\n\n"
  "El nivel m\\'aximo al que se ha llegado "
  "(el valor de la tabla) es $%d$.\n\n",
  maximo_nivel_hallado);
  printf("\n\nNivel de contradiccion: %d\n\n", maximo_nivel_hallado);
}

/*************************************************************************/
void inicia_superpila()
/*************************************************************************/
/* La siguiente funcion prepara una pila para guardar nuevos datos en  */
/* la tecnica 2 de factorizacion.                                       */
/*************************************************************************/
{
  caso_actual=1;
  vermem(superpila=malloc(sizeof(PILA)));
  superpila->siguiente=(PILA*)NULL;
  superpila->equis_actuales=(casillamodp*)NULL;
  superpila->substitucion_actual=(casillamodp*)NULL;
  superpila->numero_de_caso=0;
  superpila->nivel_de_substitucion=0;
}

/*************************************************************************/
void anyade_a_superpila(casillamodp *substitucion)
/*************************************************************************/
/* Esta funcion anyade a la superpila una substitucion procedente de   */
/* una factorizacion.  Guarda el contenido de las equis actuales con   */
/* animo de poderlas restaurar despues.                                 */
/*************************************************************************/
{
  PILA *nueva_superpila;
  int i, j;

  vermem(nueva_superpila=malloc(sizeof(PILA)));
  nueva_superpila->numero_de_caso=caso_actual;

  fprintf(salida_TeX, "\nA\\~nadimos a la pila $");
  escribe_casillamodp_archivo_TeX_l(*substitucion,"x",salida_TeX);
  fprintf(salida_TeX, "$ y lo guardamos como el caso $n=%d$.\n\n",
  caso_actual);
  caso_actual++;
  vermem(nueva_superpila->equis_actuales=malloc(C_0_MAX *
sizeof(casillamodp)));
  for (i=0; i<C_0_MAX; i++)
    {
      (nueva_superpila->equis_actuales+i)->longitud=C_0_MAX;
      for (j=0; j<C_0_MAX; j++)
{
  ((nueva_superpila->equis_actuales)+i)->coef[j]=(equis+i)->coef[j];
};
    };
  nueva_superpila->nivel_de_substitucion=nivel_de_reserva;
  nueva_superpila->substitucion_actual=substitucion;
  nueva_superpila->siguiente=superpila;
  superpila=nueva_superpila;
```

```
 }


/*************************************************************************/
void restaura_superpila()
/*************************************************************************/
/* Esta funcion restaura los valores de la superpila tras haberse      */
/* producido una contradiccion.                                        */
/*************************************************************************/
{
  int i, j;
  PILA *vieja_superpila;
  casillamodp *substitucion;

  vieja_superpila=superpila->siguiente;
  fprintf(salida_TeX, "\n\nRecuperamos de la pila el valor $n=%d$.\n\n",
  superpila->numero_de_caso);
  for (i=0; i<C_O_MAX; i++)
    {
      (equis+i)->longitud=((superpila->equis_actuales)+i)->longitud;
      for (j=0; j<C_O_MAX; j++)
{
  (equis+i)->coef[j]=((superpila->equis_actuales)+i)->coef[j];
};
    };
  free(superpila->equis_actuales);
  if (vieja_superpila!=NULL)
    {
      nivel=superpila->nivel_de_substitucion;
      nivel_de_reserva=nivel;
      substitucion=superpila->substitucion_actual;
      /* Escribimos los valores validos para el algebra de Lie */
      fprintf(salida_TeX, "\\'Estos son los valores actuales en el "
      "nivel $%d$:\n"
      "\\begin{eqnarray*}\n", nivel);
      for (j=0; j<C_O_MAX; j++)
{
  fprintf(salida_TeX, "x_{%d}&=&", j+1);
  escribe_casillamodp_archivo_TeX_1(*(equis+j), "x",
    salida_TeX);
  if (j<C_O_MAX-1)
    {
      fprintf(salida_TeX, "\\\\\n");
    };
};
      fprintf(salida_TeX, "\n\\end{eqnarray*}\n\n");
      fflush(salida_TeX);


      fprintf(salida_TeX, "Substituci\\'on actual:\n$$");
      escribe_casillamodp_archivo_TeX_1(*substitucion, "x", salida_TeX);
      fprintf(salida_TeX, "$$\n\n");
      substituir_formas_modp(*substitucion);
      fprintf(salida_TeX, "\\'Estos son los valores tras la "
      "substituci\\'on en el nivel $%d$:\n"
      "\\begin{eqnarray*}\n", nivel);
      for (j=0; j<C_O_MAX; j++)
```

```
{
  fprintf(salida_TeX, "x_{%d}&=&", j+1);
  escribe_casillamodp_archivo_TeX_1(*(equis+j), "x",
    salida_TeX);
  if (j<C_0_MAX-1)
    {
      fprintf(salida_TeX, "\\\\\n");
    };
};
      fprintf(salida_TeX, "\n\\end{eqnarray*}\n\n");
      fflush(salida_TeX);

    };
  free(superpila);
  superpila=vieja_superpila;
}

/************************************************************************/
int aplica_tecnica_2_modp(gran_jacobi_modp *jac, int n)
/************************************************************************/
/* Esta funcion intenta factorizar Jacobis que no son de la forma      */
/* $x_lt(x_1,\ldots, x_{p-1})$.                                        */
/*                                                                      */
/************************************************************************/
{
  int a, b, j;
  int contradiccion;
  casillamodp *forma1, *forma2, alp, alp1, alp2;

  fprintf(salida_TeX, "\n\nNivel %d\n\nT\\'ecnica de "
  "factorizaci\\'on 2\n", n);
  se_ha_despejado=0;
  contradiccion=0;
  vermem(forma1=malloc(sizeof(casillamodp)));
  vermem(forma2=malloc(sizeof(casillamodp)));
  if ((n==primo+2*l-c0+1) && (l>1))
    {
      /* Podemos aplicar argumentos con las z_i */
#ifdef REQUETEVERBOSE
      printf("Intentamos aplicar argumentos de zetas....\n\n");
#endif
      for (j=1; j<2*l; j++)
{
  if ((j==l-1) || (j==l))
    {
      continue;
    };
  alp=alpha_modp(j, primo-c0+1);
  if (!es_cero_cas_modp(alp))
    {
      substituir_formas_modp(alp);
#ifdef REQUETEVERBOSE
      printf("Substitucion $z_{%d}=z_{%d}$.\n", j, j+1);
#endif
      se_ha_despejado=1;
    };
};
```

```
      alp1=alpha_modp(l-1, primo-c0+1);
      alp2=alpha_modp(l, primo-c0+1);
      alp=sum_cas_modp(alp1, alp2);
      if (!es_cero_cas_modp(alp))
{
  substituir_formas_modp(alp);
#ifdef REQUETEVERBOSE
  printf("Substitucion $z_{%d}=z_{%d}$.\n", l-1, l+1);
#endif
  se_ha_despejado=1;
};
    };
  for (a=(n/3)-1; a>0 && !contradiccion && !se_ha_despejado; a--)
    {
      for (b=(n-a-1)/2; b>a && !contradiccion && !se_ha_despejado;
   b--)
{
  jacobi_modp(a,b,n-a-b, jac);
#ifdef REQUETEVERBOSE
  printf("Jacobi (%d, %d, %d):\n", a, b, n-a-b);
  escribe_jacobi_modp(*jac);
#endif
  if (!es_cero_jacobi_modp(*jac))
    {
      se_ha_despejado=factorizar(*jac,primo,forma1, forma2);
      if (se_ha_despejado)
{
  hay_jacobi_despejado=1;
  if (son_casillamodps_iguales(*forma1, *forma2))
    {
#ifdef REQUETEVERBOSE
      printf("** Factorizacion tipo 2 -"
     " Cuadrado **\n");
#endif
      fprintf(salida_TeX, "\nSe ha producido la"
      " factorizaci\\'on $(");
      escribe_casillamodp_archivo_TeX_l(*forma1, "x",
salida_TeX);
      fprintf(salida_TeX, ")^2$\nal considerar"
      " Jacobi para\n$(%d,%d,%d)$.", a, b,
      n-a-b);
    }
  else
    {
#ifdef REQUETEVERBOSE
      printf("** Factorizacion tipo 2 -"
     " No cuadrado **\n");
#endif
      fprintf(salida_TeX, "\nSe ha producido la"
      " factorizaci\\'on $(");
      escribe_casillamodp_archivo_TeX_l(*forma1,"x",
salida_TeX);
      fprintf(salida_TeX, ")(");
      escribe_casillamodp_archivo_TeX_l(*forma2,"x",
salida_TeX);
      fprintf(salida_TeX, ")$\nal considerar Jacobi"
      " para\n$(%d,%d,%d)$."
```

```
      "\n\nNos quedamos con"
      " el primer factor.\n", a, b, n-a-b);
      anyade_a_superpila(forma2);
    };
  contradiccion=substituir_formas_modp(*forma1);
  fprintf(salida_TeX, "\nConsideremos "
  "$0=\\bigl(");
  escribe_casillamodp_archivo_TeX_l(*forma1, "x",
    salida_TeX);
  fprintf(salida_TeX, "\\bigr)$.\n\n");

}
#ifdef LINEAL
      else
{
  procesar_jacobi_no_fact(*jac);
}
#endif
      ;
    };
};
    };
  /* Escribimos los valores validos para el algebra de Lie */
  fprintf(salida_TeX, "\\begin{eqnarray*}\n");
  for (j=0; j<C_0_MAX; j++)
    {
      fprintf(salida_TeX, "x_{%d}&=&", j+1);
      escribe_casillamodp_archivo_TeX_l(*(equis+j), "x",
salida_TeX);
      if (j<C_0_MAX-1)
{
  fprintf(salida_TeX, "\\\\\n");
};
    };
  fprintf(salida_TeX, "\n\\end{eqnarray*}\n\n");
  fflush(salida_TeX);
  return (contradiccion);
}
```

A.2.9. El archivo `pjacobi.h`

```
/*********************************************************************/
/*********************************************************************/
/* Archivo pjacobi.h                                                 */
/* Contiene las declaraciones de las funciones, macros y variables   */
/* globales del programa pjacobi de obtencion de cotas del grado de  */
/* conmutatividad de un $p$-grupo de clase maximal en funcion de los */
/* parametros $c_0$ y $l$.                                           */
/*********************************************************************/
/*********************************************************************/

/*********************************************************************/
/*********************************************************************/
/**           DEFINICIONES DE MACROS DE CARACTER GENERAL            **/
```

```
/*********************************************************************/
/*********************************************************************/

/*********************************************************************/
/* #define DRAFT 1                                                   */
/* #define DRAFTSHANKS 1                                             */
/* #define REQUETEVERBOSE 1                                          */
#define SIN_SALIDA 1
/* #define LINEAL 1                                                  */

/*********************************************************************/
/* DRAFT debe ser definido para escribir en la salida estandar los   */
/* calculos adicionales de la factorizacion.                         */
/*********************************************************************/
/* DRAFTSHANKS debe ser definido para compilar escribiendo todos los */
/* calculos de los Jacobi/Legendre                                   */
/*********************************************************************/
/* REQUETEVERBOSE debe ser definido para que se escriban todos los   */
/* Jacobis y se vea la marcha de la factorizacion.                   */
/*********************************************************************/
/* SIN_SALIDA debe ser definido para las pruebas.                    */
/*********************************************************************/
/* LINEAL debe ser definido para usar lineal.c                       */
/*********************************************************************/


#define COEFMAX 47
#define MAX_STACK_OVERFLOW 1024
#define MAXJAC 100

/*********************************************************************/
/*********************************************************************/
/**              DEFINICIONES DE MACROS-FUNCIONES                   **/
/*********************************************************************/
/*********************************************************************/

/*********************************************************************/
/* La macro vermem() verifica que el argumento, que debe ser un      */
/* puntero definido mediante malloc() o alguna funcion similar,      */
/* define efectivamente una nueva zona de memoria.  En caso          */
/* contrario, interrumpe el programa.                                */
/*********************************************************************/

#define vermem(xx) if ((xx)==NULL) {\
      fprintf(stderr,"Error de"\
      " memoria\n");\
exit(1);\
}

/*********************************************************************/
/*********************************************************************/
/**                ABREVIATURAS PARA ESTRUCTURAS                    **/
/*********************************************************************/
/*********************************************************************/

#define PILA struct pila
#define EUCLIDES struct euclides
```

```
/*********************************************************************/
/*********************************************************************/
/**             DEFINICIONES DE TIPOS: ESTRUCTURAS                  **/
/*********************************************************************/
/*********************************************************************/

/*********************************************************************/
/* El tipo casillamodp corresponde a una forma lineal en las $x_i$,  */
/* que se colocara como casilla en la matriz $T_G$.                  */
/*********************************************************************/
typedef struct
{
  unsigned char longitud;
  long int coef[COEFMAX];
} casillamodp;

/*********************************************************************/
/* El tipo despejemodp supone una representacion de la manera de     */
/* despejar una variable, situada en pos, de la casillamodp cas.     */
/* Esta funcion, aunque bastante trivial, corresponde a la funcion   */
/* despeje de otros programas que trabajan con elementos en los que  */
/* el elemento que hay que despejar no es el ultimo.                 */
/*********************************************************************/

typedef struct
{
  casillamodp cas;
  int pos;
} despejemodp;

/*********************************************************************/
/* La estructura euclides es una manera de representar el maximo     */
/* comun divisor de dos numeros enteros, junto con los coeficientes  */
/* lambda y mu del teorema de Bezout.  El nombre de esta estructura  */
/* corresponde al algoritmo utilizado para calcular dicho mcd (que   */
/* se almacena en g).                                                */
/*********************************************************************/

struct euclides
{
  long int g;
  long int lambda;
  long int mu;
};

/*********************************************************************/
/* La estructura psolucion representa las soluciones de una ecuacion */
/* de segundo grado en $Z_p$.  El campo haysolucion indica si la     */
/* ecuacion es resoluble o no lo es.                                 */
/*********************************************************************/

typedef struct
{
  int haysolucion;
  long int sol1;
  long int sol2;
```

```
} psolucion;

/*********************************************************************/
/* El tipo de datos pila permite almacenar uno de los valores de las */
/* substituciones en las disyuntivas que se plantean al factorizar   */
/* los Jacobi cuando uno de los factores no es $x_l$.  Almacena un   */
/* numero de caso para las referencias, el valor de las $x_i$        */
/* actuales, el nivel para el cual estamos seguros que todas las     */
/* ecuaciones de Jacobi de niveles no superiores a el se satisfacen, */
/* la forma que se despeja y el valor siguiente para ser recuperado. */
/*********************************************************************/

struct pila
{
  int numero_de_caso;
  casillamodp *equis_actuales;
  int nivel_de_substitucion;
  casillamodp *substitucion_actual;
  PILA *siguiente;
};

/*********************************************************************/
/* El puntero gran_jacobi_modp sirve para almacenar los valores de   */
/* $x_{i+l}$ en funcion de las variables que queden libres.  Tambien */
/* se usa para variables auxiliares de calculo con datos de este     */
/* tipo.                                                             */
/*********************************************************************/

typedef long int **gran_jacobi_modp;


/*********************************************************************/
/* solucion no es mas que un puntero a una psolucion, solucion de    */
/* una ecuacion de segundo grado.                                    */
/*********************************************************************/

typedef psolucion *solucion;


/*********************************************************************/
/*********************************************************************/
/**                       VARIABLES GLOBALES                        **/
/*********************************************************************/
/*********************************************************************/
PILA *superpila;


casillamodp *equis;


long int C_0_MAX, primo, ene, terminos_asignados;


int l, c0, STACK_OVERFLOW, caso_actual, nivel, maximo_nivel_hallado,
  nivel_de_reserva, contradiccion, se_ha_despejado, hay_que_repetir,
  hay_jacobi_despejado, num_jacobis_en_pila;
```

```
FILE *salida, *salida_TeX, *salida_cab_TeX, *salida_maple;


/*********************************************************************/
/*********************************************************************/
/**                       FUNCIONES DEFINIDAS                       **/
/*********************************************************************/
/*********************************************************************/


/*********************************************************************/
/*                             power.c                               */
/*********************************************************************/


long int power(long int base, long int exponente, long int modulo);
/*********************************************************************/
/* Esta funcion calcula la potencia base^exponente y reduce el       */
/* resultado al modulo dado.  Ha sido extraida del libro de Peter    */
/* Giblin, "Primes and Programming".                                 */
/*********************************************************************/




/*********************************************************************/
/*                           legendre.c                              */
/*********************************************************************/

void WriteRatio(long int n, long int k, int sign);
/*********************************************************************/
/* Esta funcion escribe en pantalla de una manera simplificada los   */
/* simbolos de Legendre/Jacobi que aparecen en el calculo.  Solo se  */
/* usa cuando esta definido DRAFTSHANKS                              */
/*********************************************************************/

int legendre(long int n, long int k);
/*********************************************************************/
/* Esta funcion devuelve el simbolo de Legendre/Jacobi ${n\brack     */
/* k}$. La idea del algoritmo es del texto de Giblin.                */
/*********************************************************************/




/*********************************************************************/
/*                             shanks.c                              */
/*********************************************************************/

long int shanks_sqrt_intermedio(long int a,long int p,long int z);
/*********************************************************************/
/* La funcion shanks_sqrt_intermedio determina la raiz cuadrada de a */
/* dado el valor inicial z del algoritmo de Shanks.                  */
/*********************************************************************/
```

```
long int shanks_inicial(long int p);
/*********************************************************************/
/* Obtiene un valor inicial para el algoritmo de Shanks, esto es, un */
/* no residuo cuadratico.                                            */
/*********************************************************************/

long int shanks_sqrt(long int a, long int p);
/*********************************************************************/
/* Algoritmo de Shanks para el calculo de la raiz cuadrada mod p     */
/*********************************************************************/




/*********************************************************************/
/*                              modular.c                            */
/*********************************************************************/


long int choose_modp(unsigned long int x1, unsigned long int x2);
/*********************************************************************/
/* Esta funcion calcula el numero combinatorio ${x1\choose x2}$ mod  */
/* p.                                                                */
/*********************************************************************/

EUCLIDES eucgcd(long int a0, long int b0);
/*********************************************************************/
/* Esta funcion calcula el maximo comun divisor de dos numeros       */
/* enteros, a0 y b0, asi como los coeficientes que aparecen en la    */
/* relacion de Bezout.  Estos tres datos aparecen reflejados en el   */
/* tipo EUCLIDES.  El algoritmo empleado es el de Euclides, en la    */
/* version descrita en lenguaje PASCAL por Peter Gibling, "Primes    */
/* and Programming", Cambridge, en la pagina 21.                     */
/*********************************************************************/

long int invmodp(long int a0, long int p);
/*********************************************************************/
/* En esta funcion se obtiene el inverso de a0 mod p. Hace uso de la */
/* funcion definida anteriormente, y, en el caso de que p no sea     */
/* primo y $\gcd(a0, p)\ne 1$, se genera un mensaje de error.        */
/*********************************************************************/

casillamodp substituir_modp(casillamodp, despejemodp);
/*********************************************************************/
/* Substituye en una casillamodp una variable despejada mod p de     */
/* otra casilla.                                                     */
/*********************************************************************/

despejemodp despejar_una_variablemodp(casillamodp c);
/*********************************************************************/
/* En esta funcion se intenta despejar una variable (mod p) de una   */
/* casilla.  Despejamos la variable situada mas a la derecha.        */
/*********************************************************************/

casillamodp prodesc_modp(long int, casillamodp);
```

```
/*********************************************************************/
/* En esta funcion se da el resultado de multiplicar un escalar (un  */
/* entero mod p) por una casilla (esto es, una forma lineal).        */
/*********************************************************************/

/*********************************************************************/
casillamodp menos_cas_modp(casillamodp cas);
/*********************************************************************/
/* Multiplica por -1 la casillamodp cas.                             */
/*********************************************************************/

casillamodp sum_cas_modp(casillamodp, casillamodp);
/*********************************************************************/
/* Suma dos casillas mod p.                                          */
/*********************************************************************/

int es_cero_cas_modp(casillamodp);
/*********************************************************************/
/* Devuelve 1 si la casilla es nula, 0 en caso contrario.            */
/*********************************************************************/

void escribe_casillamodp(casillamodp);
/*********************************************************************/
/* Escribe en stdout una casilla mod p.                              */
/*********************************************************************/

/*********************************************************************/
int son_casillamodps_iguales(casillamodp cas1, casillamodp cas2);
/*********************************************************************/
/* Devuelve $1$ si las dos casillas son iguales, $0$ en otro caso.   */
/*********************************************************************/

casillamodp alpha_modp(int i, int j);
/*********************************************************************/
/* Calcula $\alpha_{i,j}$ mod p en funcion de los $x_k$.             */
/*********************************************************************/

void inicia_equis_modp(void);
/*********************************************************************/
/* Inicializa el vector de las $x_i$, haciendo cada elemento igual a */
/* la $x_i$ correspondiente, y aplicando despues, mientras sea       */
/* posible, la periodicidad mod p-1.                                 */
/*********************************************************************/

int factorizable_l_modp(gran_jacobi_modp);
/*********************************************************************/
/* Devuelve $1$ si la relacion de Jacobi tiene como factor comun     */
/* $x_l$, y $0$ en otro caso.                                        */
/*********************************************************************/

casillamodp entre_x_l_modp(gran_jacobi_modp);
/*********************************************************************/
/* Trata de dividir el Jacobi entre $x_l$, supuesto que $x_l$ es     */
/* factor comun.                                                     */
/*********************************************************************/

int substituir_formas_modp(casillamodp);
```

```
/*********************************************************************/
/* Dada una casilla, despeja una variable y substituye el resultado  */
/*en el vector de las $x_i$ hasta que $x_l=0$.  En este caso,        */
/*devuelve $1$, en otro caso, substituye todas las variables $x_i$ y */
/*devuelve $0$.                                                      */
/*********************************************************************/

int es_cero_jacobi_modp(gran_jacobi_modp);
/*********************************************************************/
/* Devuelve $1$ si el Jacobi correspondiente es nulo, $0$ en caso    */
/* contrario.                                                        */
/*********************************************************************/

void jacobi_modp(int i, int j, int k, gran_jacobi_modp *res);
/*********************************************************************/
/* Asigna a *res los coeficientes de la forma cuadratica f(i,j,k).   */
/*********************************************************************/

void escribe_casillamodp_archivo_TeX_l(casillamodp cas, char *nom_var,
       FILE *archivo);
/*********************************************************************/
/* Escribe en un archivo TeX el valor de una casilla, denotando con  */
/* el nombre *nom_var las diversas variables.                        */
/*********************************************************************/

void escribe_jacobi_modp(gran_jacobi_modp jac);
/*********************************************************************/
/* Escribe en pantalla el valor de un Jacobi.                        */
/*********************************************************************/

void verifica(int valor);
/*********************************************************************/
/* Esta funcion detiene el programa si su argumento, una funcion de  */
/* escritura, devuelve un numero negativo.                           */
/*********************************************************************/




/*********************************************************************/
/*                            jacarbol.c                             */
/*********************************************************************/

int main_function(int argc, char *argv[]);
/*********************************************************************/
/* Funcion principal de este programa.  Contiene la descripcion      */
/* general del algoritmo.                                            */
/*********************************************************************/

void aplica_jacobi_modp(gran_jacobi_modp *jac);
/*********************************************************************/
/* Aplica las relaciones de Jacobi para los distintos niveles        */
/* tratando de factorizarlas hasta  conseguir una contradiccion.     */
/*********************************************************************/

void inicia_superpila();
```

```
/*********************************************************************/
/* Esta funcion prepara una pila para guardar nuevos datos en        */
/* la tecnica 2 de factorizacion.                                    */
/*********************************************************************/

void anyade_a_superpila(casillamodp *substitucion);
/*********************************************************************/
/* Esta funcion anyade a la superpila una substitucion procedente de */
/* una factorizacion.  Guarda el contenido de las equis actuales con */
/* animo de poderlas restaurar despues.                              */
/*********************************************************************/

void restaura_superpila();
/*********************************************************************/
/* Esta funcion restaura los valores de la superpila tras haberse    */
/* producido una contradiccion.                                      */
/*********************************************************************/

int aplica_tecnica_2_modp(gran_jacobi_modp *jac, int n);
/*********************************************************************/
/* Esta funcion intenta factorizar Jacobis que no son de la forma    */
/* $x_lt(x_l,\ldots, x_{p-1})$.                                      */
/*********************************************************************/
```

```
/*********************************************************************/
/*                              quadr.c                              */
/*********************************************************************/


void solve2(long int a, long int b, long int c, long int p, solucion
    res);
/*********************************************************************/
/* Esta funcion obtiene la solucion de la ecuacion de segundo grado  */
/* $ax^2+bx+c=0$ en el cuerpo $Z_p$ (supuesto que sea resoluble) y   */
/* coloca la solucion en res.  Caso de no ser resoluble, devuelve un */
/* puntero con un indicador de que no hay solucion.                  */
/*********************************************************************/
```

```
/*********************************************************************/
/*                          factors.c                                */
/*********************************************************************/

int factorizar(gran_jacobi_modp a, long int p, casillamodp *b,
       casillamodp *c);
/*********************************************************************/
/* Trata de dar una factorizacion de la expresion de Jacobi a mod p. */
/* Caso de existir, la coloca en los punteros b y c.  Devuelve un    */
```

```
/* entero que indica si dicha solucion existe o no existe.          */
/*******************************************************************/
```

```
/*******************************************************************/
/*                            pjacobi.c                            */
/*******************************************************************/

int main(int argc, char *argv[]);
/*******************************************************************/
/* Funcion main() que no hace mas que llamar a main_function().    */
/*******************************************************************/
```

A.2.10. El archivo `Makefile`

```
cc=cc -Wall
OBJETOS=arbol.o despeje.o fraccion.o casilla.o  primos.o  triang.o output.o
OBJETOSVL=$(OBJETOS) despejevl.o fraccionvl.o casillavl.o  primosvl.o\
 gros.o outputvl.o zetavl.o zeta2.o
cabeceras= bullet.h
all:    bullet zeta prueba

arbol.o: arbol.c  $(cabeceras)
$(cc) -c  -g arbol.c

despeje.o: despeje.c  $(cabeceras)
$(cc) -c  -g despeje.c

fraccion.o: fraccion.c  $(cabeceras)
$(cc) -c  -g fraccion.c

casilla.o: casilla.c   $(cabeceras)
$(cc) -c  -g casilla.c

output.o:  output.c $(cabeceras)
$(cc) -c  -g output.c

primos.o:  primos.c $(cabeceras)
$(cc) -c  -g primos.c

triang.o:  triang.c $(cabeceras)
$(cc) -c  -g triang.c

bullet.o: bullet.c $(cabeceras)
$(cc) -c  -g bullet.c
```

```
bullet: bullet.o $(OBJETOS) $(cabeceras)
$(cc) -o bullet  -g bullet.o $(OBJETOSVL)

zeta: zeta.o zeta2.o $(OBJETOSVL) $(cabeceras)
$(cc) -o zeta -g zeta.o $(OBJETOSVL)

zeta.o: zeta.c $(cabeceras)
$(cc) -c -g zeta.c

zeta2.o: zeta2.c $(cabeceras)
$(cc) -c -g zeta2.c

gros.o: gros.c bulletvl.h
$(cc) -c -g gros.c

fraccionvl.o: fraccionvl.c bulletvl.h
$(cc) -c -g fraccionvl.c


casillavl.o: casillavl.c bulletvl.h
$(cc) -c -g casillavl.c

zetavl.o: zetavl.c bulletvl.h
$(cc) -c -g zetavl.c

outputvl.o: outputvl.c bulletvl.h
$(cc) -c -g outputvl.c

primosvl.o: primosvl.c bulletvl.h
$(cc) -c -g primosvl.c

despejevl.o: despejevl.c bulletvl.h
$(cc) -c -g despejevl.c

prueba.o: prueba.c bulletvl.h bullet.h
$(cc) -g -c prueba.c

prueba: prueba.o $(OBJETOSVL)
$(cc) -g -o prueba prueba.o  $(OBJETOSVL)
```

Bibliografía

[1] B. Beisiegel. Semiextraspezielle p-gruppen. *Math. Z.*, 156:247–254, 1977.

[2] N. Blackburn. On a special class of p-groups. *Acta Math.*, 100:45–92, 1958.

[3] W. Burnside. *Theory of Groups of Finite Order*. Cambridge University Press. Reprinted by Dover 1955, New York, 2nd edition, 1911.

[4] M. Cartwright. Class and breadth of a finite p-group. *Bull. London Math. Soc.*, 19:425–430, 1987.

[5] B.W. Char, K.O. Geddes, G.H. Gonnet, M.B. Monagan, and S.M. Watt. *Maple reference manual*. New York, 1988.

[6] Waltraud Felsch, Joachim Neubüser, and Wilhelm Plesken. Space groups and groups of prime-power order IV. Counterexamples to the class-breadth conjecture. *J. London Math. Soc. (2)*, 24:113–122, 1981.

[7] G. A. Fernández-Alcober. The exact lower bound for the degree of commutativity of a p-group of maximal class. *J. Algebra*, 174:523–530, 1995.

[8] Joseph A. Gallian. On the breadth of a finite p-group. *Math. Z.*, 126:224–226, 1972.

[9] Peter Giblin. *Primes and Programming*. Cambridge University Press, Cambridge, Great Britain, 1993.

[10] P. Hall. A contribution to the theory of groups of prime-power order. *Proc. London Math. Soc.*, 36:29–95, 1933.

[11] P. Hall. The Eulerian functions of a group. *Quart. J. Math.*, 7:134–151, 1936.

[12] B. Huppert. *Endliche Gruppen I*, volume 134 of *Grundlehren Math. Wiss.* Springer-Verlag, Berlin, Heidelberg, New York, 1967.

[13] M. R. Jones and J. Wiegold. Isoclinism and covering groups. *Bull. Austral. Math. Soc.*, 2:71–76, 1974.

[14] C. R. Leedham-Green and Susan McKay. On p-groups of maximal class I. *Quart. J. Math. , Ser.2*, 27:297–311, 1976.

[15] C. R. Leedham-Green and Susan McKay. On p-groups of maximal class II. *Quart. J. Math.*, 29:175–186, 1978.

[16] C. R. Leedham-Green and Susan McKay. On p-groups of maximal class III. *Quart. J. Math.*, 29:281–299, 1978.

[17] L. Leedham-Green, P. M. Neumann, and James Wiegold. The breadth and the class of a finite p-group. *J. London Math. Soc*, 1:409–420, 1969.

[18] R. J. Miech. Metabelian p-groups of maximal class. *Trans. Amer. Math. Soc.*, 152:331–373, 1970.

[19] M.F. Newman. Determination of groups of prime-power order. In *Group Theory*, volume 573 of *Lecture Notes in Math.*, pages 73–84, Berlin, Heidelberg, New York, 1977. (Canberra, 1975), Springer-Verlag.

[20] E.A. O'Brien. The p-group generation algorithm. *J. Symbolic Comput.*, 9:677–698, 1990.

[21] Martin Schönert *et al.* GAP – *Groups, Algorithms and Programming.* Lehrstuhl D für Mathematik, RWTH, Aachen, 1994.

[22] R. Shepherd. *p-Groups of Maximal Class.* PhD thesis, University of Chicago, 1970. Ph. D. Thesis, University of Chicago, 1970.

[23] M. R. Vaughan-Lee. Breadth and commutator subgroups of p-groups. *Journal of Algebra*, 32:278–285, 1974.

[24] A. Vera López. Arithmetical conditions on the conjugacy vector of a finite group. *Isr. J. Math.*, 56:179–187, 1986.

[25] A. Vera-López, J. M. Arregi, M. A. García-Sánchez, F. J. Vera-López, and R. Esteban-Romero. The exact bounds for the degree of commutativity of a p-group of maximal class, I. to appear.

[26] A. Vera-López, J. M. Arregi, M. A. García-Sánchez, F. J. Vera-López, and R. Esteban-Romero. The exact bounds for the degree of commutativity of a p-group of maximal class, II. to appear.

[27] A. Vera-López, J. M. Arregi, M. A. García-Sánchez, F. J. Vera-López, and R. Esteban-Romero. Some algorithms for the computation of bounds for the degree of commutativity of a p-group of maximal class. to appear.

[28] A. Vera-López, J. M. Arregi, and F. J. Vera-López. Some bounds for the degree of commutativity of a p-group of maximal class, III. *to appear in Cambridge J. of Math.*

[29] A. Vera-López, J. M. Arregi, and F. J. Vera-López. Some bounds for the degree of commutativity of a p-group of maximal class, IV. Sin publicar.

[30] A. Vera-López, J. M. Arregi, and F. J. Vera-López. Some bounds for the degree of commutativity of a p-group of maximal class, II. *Comm. in Algebra*, 23(7):77–116, 1995.

[31] A. Vera López and G. A. Fernández-Alcober. Some bounds for the degree of commutativity of a p-group of maximal class, ii. *Comm. in Algebra*, 53(7):2765–2795, 1995.

[32] A. Vera López and Gustavo A. Fernández Alcober. The conjugacy vector of a p-group of maximal class. *Israel Journal of Mathematics*, 86:233–252, 1994.

[33] A. Vera López and M. C. Larrea. On the number of conjugacy classes in a finite p-group. *Archiv der Mathematik*, 53:126–133, 1989.

[34] A. Vera López and M. C. Larrea. On p-groups of maximal class. *Journal of Algebra*, 137:77–116, 1991.

[35] Antonio Vera López. Clases de conjugación de cardinalidad máxima en un p-grupo finito. Artículo sin publicar, 1995.

[36] Libero Verardi. On groups whose noncentral elements have the same finite number of conjugates. *Bolletino U.M.I.*, 7(2-A):391–400, 1988.

[37] A. Wiman. über mit Diedergruppen verwandte p-Gruppen. *Arkiv för Matematik, Astronomi och Fysik*, 33, 1946. vol 33A.

Índice alfabético