CAPÍTULO 5

EL SEGUNDO TEOREMA GENERAL DE LA FACTORIZACIÓN DE CORDERO ALGORITMOS Y PROGRAMAS COMPUTACIONALES

Fecha de aceptación: 01/07/2024

Ronald Cordero Méndez
Universidad Internacional San Isidro Labrador

Juan Francisco Gamboa Abarca
Universidad Internacional San Isidro Labrador

RESUMEN: En el artículo se publica El Segundo Teorema General de La Factorización de Cordero, en el Conjunto de Los Números Enteros, los Algoritmos de Cordero y su aplicación en los software construidos a partir de estos algoritmos. El Segundo Teorema General de La Factorización de Cordero permite factorizar en dos factores, los números polinomiales de la forma $4r^2n^2 + 4rn + 2pr^2 + 1 = (2rn + 1)^2 + 2pr^2$, con $r \in Z$, $n \in Q$, $r \neq 0$ y $p \in \{3, 5, 11, 29\}$. Los Algoritmos de Cordero se utilizan para factorizar en sus factores primos estos números polinomiales y los software construidos a partir del Teorema y los algoritmos nos permiten verificar tales descubrimientos matemáticos.
PALABRAS CLAVE: Factorización, Números enteros, Algoritmos, Programas computacionales, Software, Teorema

ABSTRACT: The article publishes The Second General Theorem of Cordero Factorization, in the Set of Integers, Cordero's Algorithms and their application in software built from these Algorithms. The Second General Theorem of Cordero Factorization allows factoring into two factors, the polynomial numbers of the form $4r^2n^2 + 4rn + 2pr^2 + 1 = (2rn + 1)^2 + 2pr^2$, with $r \in Z$, $n \in Q$, $r \neq 0$ y $p \in \{3, 5, 11, 29\}$. Cordero's Algorithms are used to factorize these polynomial numbers into their prime factors and the software built from the Theorem and the algorithms allow us to verify such mathematical discoveries.
KEYWORDS: Factorization, Integers, Algorithms, Computer programs, Software, Theorem.

INTRODUCCIÓN

La factorización de un número entero es un problema no resuelto en tiempo polinómico en un ordenador clásico. Es un problema sin resolver en la ciencia informática. Cuando los números enteros son muy grandes, no hay ningún algoritmo eficiente de factorización de enteros no cuántico. Tampoco se puede descartar que algún día se pueda descubrir un algoritmo computacionalmente manejable que pueda

factorizar de un vez por todas un número entero muy grande. Es la tremenda dificultad que tiene la factorización de números enteros que permite ser utilizada en los algoritmos empleados en criptografía, como es el caso del cifrado de clave pública RSA y la firma digital RSA. Se han utilizado muchas áreas de la matemática y la informática a tan complicado problema, como las curvas elípticas, la teoría algebraica de números, la criba general del cuerpo de números y recientemente la computación cuántica.

No todos los números de una determinada longitud son igualmente difíciles de factorizar. Los casos más difíciles de estos problemas (para las técnicas actualmente conocidas) son los semiprimos, el producto de dos números primos. Cuando ambos son grandes, por ejemplo, más de dos mil bits de largo, elegidos al azar y aproximadamente del mismo tamaño (pero no demasiado cerca, por ejemplo, para evitar la factorización eficiente por el método de factorización de Fermat), los algoritmos de factorización en las computadoras más rápidas pueden tomar suficiente tiempo para hacer que la búsqueda no sea práctica; es decir, a medida que aumenta la cantidad de dígitos de los números compuestos que se factorizan, la cantidad de operaciones requeridas para realizar la factorización en cualquier computadora aumenta drásticamente.

Todavía no se ha publicado un algoritmo que pueda factorizar todos los enteros en tiempo polinomial, es decir, que se pueda factorizar un número de $b - bit\ n$ en tiempo $O(b^k)$ para alguna constante k. Hasta el momento no se sabe si el problema es de clase NP o no es NP- completo.

EL SEGUNDO TEOREMA GENERAL DE LA FACTORIZACIÓN DE CORDERO

El teorema que se publica en este artículo tiene como objetivo factorizar números enteros de la forma $4r^2n^2 + 4rn + 2pr^2 + 1 = (2rn + 1)^2 + 2pr^2$, con $r \epsilon Z$, $n \epsilon Q$, $r \neq 0$ y $p \epsilon \{3,5,11,29\}$.

SEGUNDO TEOREMA GENERAL DE LA FACTORIZACIÓN DE CORDERO.

Sea r, x, $T \epsilon Z$, $r \neq 0$ y $p \epsilon \{3,5,11,29\}$

I. Si $n = (2rx + 1)^2 + 2pr^2) * T + 4rx^2 + 3x + 2pr$ entonces $(2rn + 1)^2 + 2pr^2$ es un número compuesto y dos de sus factores tienen la forma:

$$f_1 = (2rx + 1)^2 + 2pr^2$$
$$f_2 = [4r(rT+1)x + (2rT+1)]^2 + 8pr^2(rT+1)^2$$

II. Si $n = (2r^2x^2 + p) * T + \frac{2r^2x^2+2r^2x+p-1}{2r}$ entonces $(2rn + 1)^2 + 2pr^2$ es un número compuesto y dos de sus factores tienen la forma:

$$f_1 = 2r^2x^2 + p$$
$$f_2 = 2r^2[(2rT + 1)x + 1]^2 + p(2rT + 1)^2$$

III. Si $n = (2r^2x^2 + p) * T - (\frac{2r^2x^2+2r^2x+p+1}{2r})$ entonces $(2rn + 1)^2 + 2pr^2$ es un número compuesto y dos de sus factores tienen la forma:

$$f_1 = 2r^2x^2 + p$$
$$f_2 = 2r^2[(2rT - 1)x - 1]^2 + p(2rT - 1)^2$$

Aplicación de la primera parte del Teorema

Sea $r = 8$, $p = 29$, $T = 3$, $x = 6$

$$n = ((2rx + 1)^2 + 2pr^2) * T + 4rx^2 + 3x + 2pr$$
$$n = (97^2 + 3712) * T + 1634$$
$$n = 40997$$

El número polinomial a factorizar es:

$(2rn + 1)^2 + 2pr^2 = 655953^2 + 58 * 64 = 430274341921 = 13121 * 32792801$

El factor 32792801 se puede encontrar por división (como ya conocemos el factor 13121 puesto que es parte de R). Pero también hay una fórmula para encontrarlo.

La fórmula para encontrar el factor f_2 es:

$$f_2 = [4r(rT + 1)x + (2rT + 1)]^2 + 8pr^2(rT + 1)^2$$

$r = 8$, $p = 29$, $T = 3$, $x = 6$

$$f_2 = 4849^2 + 14848 * 25^2$$
$$f_2 = 32792801$$

Aplicación de la segunda parte del Teorema.

Sea $r = 12$, $p = 29$, $x = -8$, $T = 6$

$$n = (2r^2x^2 + p) * T + \frac{2r^2x^2 + 2r^2x + p - 1}{2r}$$

$$n = 18461 * 6 + \frac{288 * 64 - 288 * 8 + 28}{2 * 12}$$

$$n = 18461 * 6 + \frac{4039}{6}$$

$$n = \frac{668635}{6}$$

Entonces $4r^2n^2 + 4r * n + 2pr^2 + 1 = (2rn + 1)^2 + 2pr^2$

$576 * (\frac{668635}{6}) + 48 * \frac{668635}{6}^2 + 58 * 144 + 1 = 18461 * 387474653$

El factor 387474653 se calculó por división, pero también se puede calcular con la fórmula:

$$f_2 = 2r^2[(2rT + 1)x + 1]^2 + p(2rT + 1)^2$$

$$r = 12, \qquad p = 29, \qquad x = -8, \qquad T = 6$$

$$f_2 = 288 * 1159^2 + 29 * 145^2$$

$$f_2 = 387474653$$

Aplicación de la tercera parte del Teorema.

Sea $r = 12$, $p = 29$, $x = -8$, $T = 6$

$$n = (2r^2x^2 + p) * T - \frac{2r^2x^2 + 2r^2x + p + 1}{2r}$$

$$n = 18461 * 6 - \frac{288 * 64 - 288 * 8 + 30}{2 * 12}$$

$$n = 18461 * 6 - \frac{2693}{4}$$

$$n = \frac{440371}{4}$$

Entonces $4r^2n^2 + 4r * n + 2pr^2 + 1 = (2rn + 1)^2 + 2pr^2$

$576 * (\frac{440371}{4})^2 + 48 * \frac{440371}{4} + 58 * 144 + 1 = 18461 * 378168221$

El factor 378168221 se calculó por división, pero también se puede calcular con la fórmula:

$$f_2 = 2r^2[(2rT - 1)x - 1]^2 + p(2rT - 1)^2$$

$$r = 12, \qquad p = 29, \; x = -8, \; T = 6$$

$$f_2 = 288 * 1145^2 + 29 * 143^2$$

$$f_2 = 378168221$$

ALGORITMO r, n, p DE CORDERO PARA COMPROBAR LA PRIMALIDAD DE LOS NÚMEROS POLINOMIALES DE LA FORMA $4r^2n^2 + 4rn + 2pr^2 + 1 = (2rn + 1)^2 + 2pr^2$

El algoritmo permite comprobar la primalidad de un número polinomial de la forma $\mathbf{4r^2n^2 + 4rn + 2pr^2 + 1 = (2rn + 1)^2 + 2pr^2}$**.** El usuario da valores enteros de entrada a las variables r, n, p y el algoritmo nos permite corroborar si el números es primo o compuesto. Su capacidad es de a lo más 20 dígitos.

Algoritmo r, n, p (Test de Primalidad)

Sea $n \in Z$, $r \in Z$, $r \neq 0$, $p \in \{3, 5, 11, 29\}$ y $T_{\emptyset} \in Z$, $T_{\emptyset} \neq 0$

$$\emptyset = \sqrt{-2pT_{\emptyset}^2 + 2(2rn + 1)T_{\emptyset} + r^2}$$

Si existe al menos un $T_{\phi} \in Z$, $T_{\emptyset} \neq 0$ tal que $\phi \in N$ entonces $\mathbf{(2rn + 1)^2 + 2pr^2}$ es un número compuesto, caso contrario $\mathbf{(2rn + 1)^2 + 2pr^2}$ es un número primo.

Se construye un software basado en el algoritmo r, n, p (Test de primalidad) anterior. Este software se basa en un programa construido por el Español y Bioinformático Roberto Reinosa Fernández. A continuación se presentan algunas ejemplos de aplicación.

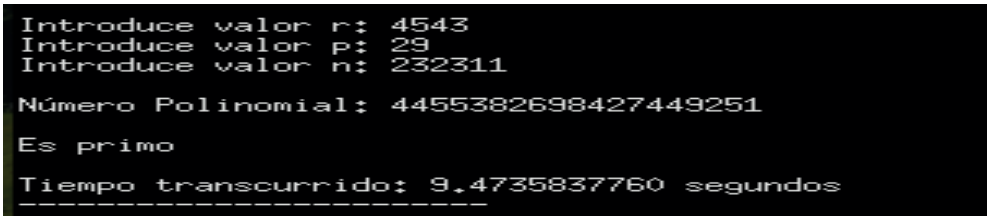
Ejemplos

1.

```
Introduce valor r: 4
Introduce valor p: 29
Introduce valor n: 78987619

Número Polinomial: 399298814402309137

Es primo

Tiempo transcurrido: 2,8566284770 segundos
--------------------------
```

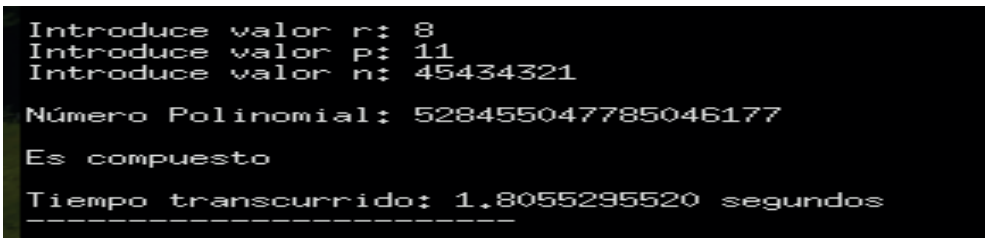
Para este número de 18 dígitos el software necesitó de menos de 3 segundos para comprobar su primalidad.

2.



El software es muy útil para encontrar números primos menores a 20 dígitos, así como números compuestos con pocos factores menores a 20 dígitos. Para este número de 19 dígitos necesitó de aproximadamente 9, 47 segundos.

3.



Para el caso de los números compuestos requiere de menos tiempo para identificarlos.

ALGORITMO r, n, p DE CORDERO PARA FACTORIZAR COMPLETAMENTE NÚMEROS DE LA FORMA $4r^2n^2 + 4rn + 2pr^2 + 1 = (2rn + 1)^2 + 2pr^2$

El algoritmo permite factorizar completamente números polinomiales de la forma $(2rn + 1)^2 + 2pr^2$. El usuario da valores de entrada r, n , p y el algoritmo construye el número polinomial $(2rn + 1)^2 + 2pr^2$, luego lo factoriza en sus factores primos. Pero también se puede utilizar el algoritmo para factorizar números polinomiales obtenidos del Segundo Teorema General de La Factorización de Cordero. Partimos de los valores de r, $s, T\ y\ x$ construimos el valor de n y luego aplicamos el algoritmo r, n, p.

Algoritmo r, n, p (Algoritmo de Cordero)

Sea $n \in Z$, $r \in Z$, $r \neq 0$, $p \in \{3, 5, 11, 29\}$ $y\ T_{\emptyset} \in Z$, $T_{\emptyset} \neq 0$

$$\emptyset = \sqrt{-2pT_{\emptyset}^2 + 2(2rn + 1)T_{\emptyset} + r^2}$$

I. Si existe al menos un $T_\phi \in Z$, $T_\varnothing \neq 0$ tal que $\phi \in N$ entonces $(\boldsymbol{2rn+1})^2 + \boldsymbol{2pr^2}$ es un número compuesto, caso contrario $(\boldsymbol{2rn+1})^2 + \boldsymbol{2pr^2}$ es un número primo.

II. Si $(\boldsymbol{2rn+1})^2 + \boldsymbol{2pr^2}$ es un número compuesto con $(T_\phi\ ,\ \phi)$, $T_\phi \in Z$, $\phi \in Z$ y $\dfrac{-\boldsymbol{r} \pm \boldsymbol{\phi}}{\boldsymbol{2} * T_\phi} = \dfrac{\boldsymbol{t}}{\boldsymbol{b}}$ con $t\ y\ b$ primos entre sí entonces 2 $t^2 + pb^2$ o $\dfrac{2\,t^2+pb^2}{2}$ es un factor de 2 $(\boldsymbol{2rn+1})^2 + \boldsymbol{2pr^2}$.

Nota: El valor de n puede pertenecer a Q, pero debe ser generado por el Segundo Teorema General de La Factorización de Cordero.

A continuación se utiliza un software construido a partir de este algoritmo. Damos algunos ejemplos.

Ejemplos

1.

```
Introduce valor r: 33
Introduce valor p: 29
Introduce valor n: 54543421

Número Polinomial: 129590336844131925 31

Factor 1: 29
Factor 2: 503791691653897
Factor 3: 887

Tiempo transcurrido: 16.33487892 segundos
--------------------------
```

Los números polinomiales siempre tienen pocos factores. Puede comprobar que 29, 887 y 503791691653897 son números primos, por lo que se trata de una factorización completa. Factoriza números polinomiales de 21 dígitos o menos.

2.

```
Introduce valor r: 4
Introduce valor p: 11
Introduce valor n: 56565432

Número Polinomial: 204777479135231201

Es primo

Tiempo transcurrido: 5.39325945 segundos
--------------------------
```

En este caso el número polinomial es primo, por lo que el software también trabaja como test de primalidad.

3.

```
Introduce valor r: 4
Introduce valor p: 29
Introduce valor n: 234343211

Número Polinomial: 351467139842408564 9

Es biprimo

Factor 1: 29287
Factor 2: 120007901062727

Tiempo transcurrido: 8,61686213 segundos
------------------------
```

En la factorización de estos números polinomiales se obtienen muchos biprimos.

```
Introduce valor r: 11
Introduce valor p: 29
Introduce valor n: 32323211

Número Polinomial: 505678346587880467

Factor 1: 241
Factor 2: 540384101
Factor 3: 3882887

Tiempo transcurrido: 3,12752069 segundos
------------------------
```

APLICACIONES DEL ALGORITMO DE CORDERO r, n, p. SEGUNDO TEOREMA GENERAL

En este apartado analizaremos la aplicabilidad del algoritmo, su importancia en la factorización completa de los números enteros polinomiales de la forma $(\mathbf{2rn + 1})^2 + \mathbf{2pr}^2$ y sus limitaciones. El algoritmo r, n, p no necesita del Teorema. Se da un valor para n entero cualquiera y sin construirlo a partir del Teorema.

Aplicación 1.

Sea $r = 7$, $n = 786$ y $p = 29$

Luego:

$$Ø = \sqrt{-2pT_Ø^2 + 2(2rn + 1)T_Ø + r^2}$$

$$Ø = \sqrt{-58 * T_Ø^2 + 22010 * T_Ø + 49}\ ,\ T_Ø \epsilon Z\ ,\ T_Ø \neq 0$$

Si aplicamos el algoritmo, obtenemos un único par ordenado entero, o sea el número polinomial es biprimo o semiprimo:

$$T_Ø = 129 \qquad y\ Ø = 1369$$

Par ordenado: (129, 1369)

Y el número polinomial es:

$$4r^2n^2 + 4rn + 2pr^2 + 1 = 4 * 49 * 786^2 + 4 * 7 * 786 + 2 * 29 * 49 + 1$$
$$= 121112867$$

Así:

$$\frac{-\boldsymbol{r} + \boldsymbol{\phi}}{\mathbf{2} * T_\phi} = \frac{\mathbf{-7 + 1369}}{\mathbf{2 * 129}} = \frac{\mathbf{227}}{\mathbf{43}}$$

Por lo que: $2 * 227^2 + 29 * 43^2 = 156679$ es un factor del número polinomial.

$$\frac{-\boldsymbol{r} - \boldsymbol{\phi}}{\mathbf{2} * T_\phi} = \frac{\mathbf{-7 - 1369}}{\mathbf{2 * 129}} = \frac{\mathbf{-16}}{\mathbf{3}}$$

Por lo que: $2 * 16^2 + 29 * 3^2 = 773$ es el otro factor del número polinomial.

Como no hay más pares ordenados enteros generados por el algoritmo, tenemos:

$$121112867 = 773 * 156679$$

Aplicación 2.

Sea $r = 8$, $n = 4099$ y $p = 11$

Luego:

$$Ø = \sqrt{-2pT_Ø^2 + 2(2rn + 1)T_Ø + r^2}$$

$$Ø = \sqrt{-22 * T_Ø^2 + 131170 * T_Ø + 64}\ ,\ T_Ø \epsilon Z\ ,\ T_Ø \neq 0$$

Si aplicamos el algoritmo, no obtenemos pares ordenados enteros. Por lo que el número polinomial:

$$4r^2n^2 + 4rn + 2pr^2 + 1 = 4 * 64 * 4099^2 + 4 * 8 * 4099 + 2 * 29 * 64 + 1$$
$$= 4301393633$$

Es un número primo.

ALGORITMO $F1\_F2$ DE CORDERO PARA FACTORIZAR EN FORMA COMPLETA LOS NÚMEROS POLINOMIALES DE LA FORMA $4r^2n^2 + 4rn + 2pr^2 + 1 = (2rn + 1)^2 + 2pr^2$ UTILIZANDO EL SEGUNDO TEOREMA GENERAL DE LA FACTORIZACIÓN DE CORDERO

Para la primera parte del Segundo Teorema General de la Factorización de Cordero tenemos que

Sea $r, x, T \epsilon Z, r \neq 0$ y $p \epsilon \{3,5,11,29\}$ $además$ $T_{\emptyset} \epsilon Z, T_{\emptyset} \neq 0$

Algoritmo para f_1

$$\emptyset = \sqrt{-2pT_{\emptyset}^2 + 2(2rn + 1)T_{\emptyset} + r^2}$$

$$f_1 = 4r^2x^2 + 4rx + 2pr^2 + 1$$

I. Si existe al menos un $T_{\phi} \epsilon Z, T_{\emptyset} \neq 0$ tal que $\phi \epsilon N$ entonces f_1 es un número compuesto, caso contrario f_1 es un número primo.

II. Si f_1 es un número compuesto con $(T_{\phi}, \phi), T_{\phi} \epsilon Z, \phi \epsilon Z$ y $\dfrac{-r \pm \phi}{2 * T_{\phi}} = \dfrac{t}{b}$ con t y b primos entre sí entonces $2t^2 + pb^2$ o $\dfrac{2t^2 + pb^2}{2}$ es un factor de f_1.

Algoritmo para f_2

$$\emptyset = \sqrt{\frac{-1}{2}pT_{\emptyset}^2 + \frac{1}{2}M T_{\emptyset} + r_*^2}$$

Donde $r_* = r(rT + 1)$ y $M = 4r_*x + 2rT + 1$

$$f_2 = 8pr_*^2 + M^2$$

III. Si existe al menos un $T_{\phi} \epsilon Z, T_{\emptyset} \neq 0$ tal que $\phi \epsilon N$ entonces f_2 es un número compuesto, caso contrario f_2 es un número primo.

IV. Si f_2 es un número compuesto con $(T_{\phi}, \phi), T_{\phi} \epsilon Z, \phi \epsilon Z$ y $\dfrac{-r_* \pm \phi}{T_{\phi}} = \dfrac{t}{b}$ con t y b primos entre sí entonces $2t^2 + pb^2$ o $\dfrac{2t^2 + pb^2}{2}$ es un factor de f_2.

Algunos ejemplos de aplicación del software construido a partir de estos algoritmos.

Ejemplo 1.

```
Introduce valor r: 56
Introduce valor p: 29
Introduce valor x: 4543
Introduce valor T: 9

Número Polinomial: 68373814313035675202936476097

Factor 1:
Factor: 1815647
Factor: 142591
Factor2
Factor: 264098708268867361

Tiempo transcurrido: 2.28153126 segundos
------------------------
```

El número polinomial de 29 dígitos no es problema para el algoritmo, el cuál lo factoriza en aproximadamente 2,29 segundos.

Ejemplo 2.

```
Introduce valor r: 3
Introduce valor p: 29
Introduce valor x: 7876541
Introduce valor T: 6

Número Polinomial: 72030160588892964034460500232792 83

Factor 1:
Factor: 23869
Factor: 93570590599
Factor2
Factor: 59
Factor: 313
Factor: 262237
Factor: 665963567

Tiempo transcurrido: 8.38769267 segundos
------------------------
```

En este otro ejemplo el número polinomial tiene 34 dígitos en solamente 8,4 segundos aproximadamente.

Ejemplo 3.

```
Introduce valor r: 12
Introduce valor p: 29
Introduce valor x: 4567843
Introduce valor T: 11

Número Polinomial: 10220047866652314148679920425877574881

Factor 1:
Factor: 1544884409
Factor: 3931
Factor: 1979
Factor2
Factor: 283117
Factor: 7949
Factor: 1394313287
Factor: 271

Tiempo transcurrido: 130.64845551 segundos
------------------------
```

Podemos considerar un número de 38 dígitos como grande, sin embargo el software lo factoriza en aproximadamente 2,18 minutos.

ALGORITMO $F3\_F4$ DE CORDERO PARA FACTORIZAR EN FORMA COMPLETA LOS NÚMEROS POLINOMIALES DE LA FORMA $4r^2n^2 + 4rn + 2pr^2 + 1 = (2rn + 1)^2 + 2pr^2$ UTILIZANDO EL SEGUNDO TEOREMA GENERAL DE LA FACTORIZACIÓN DE CORDERO

Para la segunda parte del Segundo Teorema General de la Factorización de Cordero tenemos que

Sea r, x, $T \epsilon Z$, $r \neq 0$ y $p \epsilon \{3,5,11,29\}$ $además$ $T_{\emptyset} \epsilon Z$, $T_{\emptyset} \neq 0$

Algoritmo para f_3

$$\emptyset = \sqrt{-2pT_{\emptyset}^{\,2} + 4r^2xT_{\emptyset} + r^2}$$

$$f_3 = 2r^2x^2 + p$$

I. Si existe al menos un $T_{\phi} \epsilon Z, T_{\emptyset} \neq 0$ tal que $\phi \epsilon N$ entonces f_3 es un número compuesto, caso contrario f_3 es un número primo.

II. Si f_3 es un número compuesto con (T_{ϕ}, ϕ), $T_{\phi} \epsilon Z$, $\phi \epsilon Z$ y $\dfrac{-\boldsymbol{r} \pm \boldsymbol{\phi}}{\boldsymbol{2 * T_{\phi}}} = \dfrac{\boldsymbol{t}}{\boldsymbol{b}}$ con t y b primos entre sí entonces $2t^2 + pb^2$ o $\dfrac{2t^2 + pb^2}{2}$ es un factor de f_3.

Algoritmo para f_4

$$\emptyset = \sqrt{-2pT_{\emptyset}^{\,2} + 4M\,T_{\emptyset} + r_*^{\,2}}$$

Donde $r_* = 2rT + 1$ y $M = r(r_* x + 1)$

$$f_4 = pr_*^{\,2} + 2M^2$$

III. Si existe al menos un $T_{\phi} \epsilon Z$, $T_{\emptyset} \neq 0$ tal que $\phi \epsilon N$ entonces f_4 es un número compuesto, caso contrario f_4 es un número primo.

IV. Si f_4 es un número compuesto con (T_{ϕ}, ϕ), $T_{\phi} \epsilon Z$, $\phi \epsilon Z$ y $\dfrac{-\boldsymbol{r_*} \pm \boldsymbol{\phi}}{\boldsymbol{2 * T_{\phi}}} = \dfrac{\boldsymbol{t}}{\boldsymbol{b}}$ con t y b primos entre sí entonces $2t^2 + pb^2$ o $\dfrac{2t^2 + pb^2}{2}$ es un factor de f_4.

Utilizando un programa computacional construido a partir de estos algoritmos, damos algunos ejemplos.

Ejemplos

1.

```
Introduce valor r: 4
Introduce valor p: 29
Introduce valor x: 454532
Introduce valor T: 8

Número Polinomial: 184664984882317355329305816169

Factor: 271
Factor: 24395493907
Factor: 27932232527020477

Tiempo transcurrido: 1.13498200 segundos
-------------------------
```

Utilizando el Teorema se logra factorizar números polinomiales de 30 dígitos en un tiempo de aproximadamente 1,13 segundos.

2.

```
Introduce valor r: 12
Introduce valor p: 29
Introduce valor x: 565653
Introduce valor T: 6

Número Polinomial: 178534178587462941529526945156233

Factor: 2113
Factor: 334379
Factor: 130423
Factor: 1937441920763725373

Tiempo transcurrido: 9.39699148 segundos
-------------------------
```

Se necesitó de aproximadamente 9,39 segundos para factorizar completamente el número polinomial de 33 dígitos.

3.

```
Introduce valor r: 22
Introduce valor p: 29
Introduce valor x: 2345451
Introduce valor T: 7

Número Polinomial: 270752909302166981126137338526915255 3

Factor: 7253
Factor: 734193285649
Factor: 10868862047429
Factor: 46780081

Tiempo transcurrido: 149.81176960 segundos
------------------------
```

Logra factorizar en forma completa números polinomiales de 37 dígitos, pero su tiempo es de aproximadamente 2,5 minutos.

4.

```
Introduce valor r: 6
Introduce valor p: 29
Introduce valor x: 54543421
Introduce valor T: 6

Número Polinomial: 24450075702506518674609780302760976713

Factor: 23855575663
Factor: 57191
Factor: 157
Factor: 1223687
Factor: 233
Factor: 59753299889

Tiempo transcurrido: 234.51414179 segundos
------------------------
```

Tenemos que el software logra factorizar números polinomiales de 39 dígitos y su tiempo es de aproximadamente 3,9 minutos. El software no ha sido ejecutado en un ordenador, sería interesante averiguar si en un ordenador aumenta la rapidez en el cálculo de la factorización y si logra factorizar números superiores a 39 dígitos.

ALGORITMO $F5\_F6$ DE CORDERO PARA FACTORIZAR EN FORMA COMPLETA LOS NÚMEROS POLINOMIALES DE LA FORMA $4r^2n^2 + 4rn + 2pr^2 + 1 = (2rn + 1)^2 + 2pr^2$ UTILIZANDO EL SEGUNDO TEOREMA GENERAL DE LA FACTORIZACIÓN DE CORDERO

Para la segunda parte del Segundo Teorema General de la Factorización de Cordero tenemos que

Sea r, x, $T \epsilon Z$, $r \neq 0$ y $p \epsilon \{3,5,11,29\}$ $además$ $T_{\emptyset} \epsilon Z$, $T_{\emptyset} \neq 0$

Algoritmo para f_5

$$\emptyset = \sqrt{-2pT_{\emptyset}{}^2 + 4r^2xT_{\emptyset} + r^2}$$

$$f_5 = 2r^2x^2 + p$$

V. Si existe al menos un $T_{\phi} \epsilon Z$, $T_{\emptyset} \neq 0$ tal que $\phi \epsilon N$ entonces f_3 es un número compuesto, caso contrario f_5 es un número primo.

VI. Si f_5 es un número compuesto con (T_{ϕ} , ϕ), $T_{\phi} \epsilon Z$, $\phi \epsilon Z$ y $\frac{-r \pm \phi}{2 * T_{\phi}} = \frac{t}{b}$ con t y b primos entre sí entonces $2t^2 + pb^2$ o $\frac{2t^2+pb^2}{2}$ es un factor de f_5.

Algoritmo para f_6

$$Ø = \sqrt{-2pT_Ø^2 + 4M\,T_Ø + r_*^2}$$

Donde $r_* = 2rT - 1$ y $M = r(r_* x - 1)$

$$f_6 = pr_*^2 + 2M^2$$

VII. Si existe al menos un $T_\phi \epsilon Z$, $T_Ø \neq 0$ tal que $\phi \epsilon N$ entonces f_4 es un número compuesto, caso contrario f_6 es un número primo.

VIII. Si f_6 es un número compuesto con (T_ϕ, ϕ), $T_\phi \epsilon Z$, $\phi \epsilon Z$ y $\frac{-r \pm \phi}{2 * T_\phi} = \frac{t}{b}$ con t y b primos entre sí entonces $2t^2 + pb^2$ o $\frac{2t^2+pb^2}{2}$ es un factor de f_6.

Ejemplos.

1.

```
Introduce valor r: 12
Introduce valor p: 29
Introduce valor x: 676543
Introduce valor T: 5

Número Polinomial: 24607104128152963614504174384837 7

Factor: 131820604084541
Factor: 59351
Factor: 31452065307547

Tiempo transcurrido: 11.57348470 segundos
------------------------
```

Se observa que los factores primos generalmente son números grandes. En este caso el número polinomial solamente tiene 3 factores primos grandes. El número tiene 33 dígitos y lo factoriza en menos de 12 segundos.

2.

```
Introduce valor r: 11
Introduce valor p: 29
Introduce valor x: 454543
Introduce valor T: 10

Número Polinomial: 11989990771815618896439165711202 7

Factor: 327798677
Factor: 152531
Factor: 3151148558412061
Factor: 761

Tiempo transcurrido: 12.73723677 segundos
------------------------
```

Este número polinomial tiene mayor cantidad de factores primos que el primer ejemplo, sin embargo sigue siendo pocos factores. El número polinomial tiene 33 dígitos y los factoriza en menos de 13 segundos.

3.

```
Introduce valor r: 10
Introduce valor p: 29
Introduce valor x: 2323412
Introduce valor T: 11

Número Polinomial: 559053192166094675116251218897506 01

Factor: 293
Factor: 3684807728153
Factor: 229
Factor: 423484039163
Factor: 14431
Factor: 37

Tiempo transcurrido: 62.16967999 segundos
------------------------
```

Las opciones de p son el 3, 5, 11 y 29. En este caso se utiliza p = 29, y su cantidad de factores son 6. Entre más pequeño sea p es mayor la cantidad de factores primos. Este número polinomial tiene 35 dígitos y su tiempo es de menos de 63 segundos.

4.

```
Introduce valor r: 3
Introduce valor p: 29
Introduce valor x: 67656543
Introduce valor T: 11

Número Polinomial: 28682099362276360185229031881968709003

Factor: 127
Factor: 648766461360593
Factor: 350891
Factor: 5194134940433
Factor: 191

Tiempo transcurrido: 125.59508157 segundos
------------------------
```

Se necesita de computadoras más potentes para mejorar el tiempo. Los algoritmos y softwares no han sido comprobados en ordenadores o computadoras más potentes. En este caso el número polinomial a factorizar tiene 38 dígitos pero su tiempo es de un poco más de 2 minutos. Hay que mejorar este tiempo de cálculo.

5.

```
Introduce valor r: 8
Introduce valor p: 29
Introduce valor x: 232321211
Introduce valor T: 3

Número Polinomial: 10543167504497620237889470095537699679319 3

Factor: 359
Factor: 19243906881071963
Factor: 60083
Factor: 79355629489
Factor: 3200767

Tiempo transcurrido: 941.79237627 segundos
------------------------
```

Se logran factorizar números polinomiales de 42 dígitos pero su tiempo de cálculo es de 15,7 minutos, lo cual hay que mejorar.

ALGORITMOS PARA FACTORIZAR LOS SEIS FACTORES QUE SE OBTIENEN CON EL SEGUNDO TEOREMA GENERAL DE LA FACTORIZACIÓN DE CORDERO. FACTORES $F_1 - F_2 - F_3 - F_4 - F_5 - F_6$. UN ALGORITMO PARA CADA FACTOR

Cuando utilizamos el Segundo Teorema General de la Factorización de Cordero obtenemos cinco factores diferentes, pero estos cinco factores generalmente no son números primos, por lo que utilizaremos los siguiente algoritmos para encontrar su factorización completa o factorización prima de los números polinomiales de la forma $\boldsymbol{M^2R^2 + 2pr^2(M^2 + R^2)+4p^2r^4}$ con $\boldsymbol{M = 2rn_1 + 1}$ y $\boldsymbol{R = 2rn_2 + 1}$

Algoritmos para $f_1 - f_2 - f_3 - f_4 - f_5 - f_6$ (Algoritmos de Cordero)

Sea $r, x, T\epsilon Z, r \neq 0$ y $p\epsilon\{3,5,11,29\}$ $además$ $T_{\emptyset}\epsilon Z, T_{\emptyset} \neq 0$

Algoritmo para f_1

$$\emptyset = \sqrt{-2pT_{\emptyset}^2 + 2(2rx + 1)T_{\emptyset} + r^2}$$

$$f_1 = 4r^2x^2 + 4rx + 2pr^2 + 1$$

I. Si existe al menos un $T_{\phi} \epsilon Z$, $T_{\emptyset} \neq 0$ tal que $\phi \epsilon N$ entonces f_1 es un número compuesto, caso contrario f_1 es un número primo.

II. Si f_1 es un número compuesto con (T_{ϕ}, ϕ), $T_{\phi} \epsilon Z$, $\phi \epsilon Z$ y $\frac{-\boldsymbol{r} \pm \boldsymbol{\phi}}{\boldsymbol{2} * T_{\phi}} = \frac{\boldsymbol{t}}{\boldsymbol{b}}$ con t y b primos entre sí entonces $2t^2 + pb^2$ o $\frac{2t^2+pb^2}{2}$ es un factor de f_1.

Algoritmo para f_2

$$\emptyset = \sqrt{\frac{-1}{2} p T_{\emptyset}{}^2 + \frac{1}{2} M T_{\emptyset} + r_*{}^2}$$

Donde $r_* = r(rT + 1)$ y $M = 4r_* x + 2rT + 1$

$$f_2 = 8pr_*{}^2 + M^2$$

III. Si existe al menos un $T_{\phi} \epsilon Z$, $T_{\emptyset} \neq 0$ tal que $\phi \epsilon N$ entonces f_2 es un número compuesto, caso contrario f_2 es un número primo.

IV. Si f_2 es un número compuesto con (T_{ϕ}, ϕ), $T_{\phi} \epsilon Z$, $\phi \epsilon Z$ y $\frac{-\boldsymbol{r}_* \pm \boldsymbol{\phi}}{T_{\phi}} = \frac{\boldsymbol{t}}{\boldsymbol{b}}$ con t y b primos entre sí entonces $2t^2 + pb^2$ o $\frac{2t^2+pb^2}{2}$ es un factor de f_2.

Algoritmo para f_3

$$\emptyset = \sqrt{-2pT_{\emptyset}{}^2 + 4r^2 x T_{\emptyset} + r^2}$$

$$f_3 = 2r^2x^2 + p$$

V. Si existe al menos un $T_{\phi} \epsilon Z$, $T_{\emptyset} \neq 0$ tal que $\phi \epsilon N$ entonces f_3 es un número compuesto, caso contrario f_3 es un número primo.

VI. Si f_3 es un número compuesto con (T_{ϕ}, ϕ), $T_{\phi} \epsilon Z$, $\phi \epsilon Z$ y $\frac{-\boldsymbol{r} \pm \boldsymbol{\phi}}{\boldsymbol{2} * T_{\phi}} = \frac{\boldsymbol{t}}{\boldsymbol{b}}$ con t y b primos entre sí entonces $2t^2 + pb^2$ o $\frac{2t^2+pb^2}{2}$ es un factor de f_3.

Algoritmo para f_4

$$\emptyset = \sqrt{-2pT_{\emptyset}{}^2 + 4M T_{\emptyset} + r_*{}^2}$$

Donde $r_* = 2rT + 1$ y $M = r(r_* x + 1)$

$$f_4 = pr_*^2 + 2M^2$$

VII. Si existe al menos un $T_\phi \epsilon Z$, $T_\emptyset \neq 0$ tal que $\phi \epsilon N$ entonces f_4 es un número compuesto, caso contrario f_4 es un número primo.

VIII. Si f_4 es un número compuesto con (T_ϕ, ϕ), $T_\phi \epsilon Z$, $\phi \epsilon Z$ y $\frac{-r_* \pm \phi}{T_\phi} = \frac{t}{b}$ con t y b primos entre sí entonces $2t^2 + pb^2$ o $\frac{2t^2+pb^2}{2}$ es un factor de f_4.

Algoritmo para f_5

$$\emptyset = \sqrt{-2pT_\emptyset^2 + 4r^2(x+a)T_\emptyset + r^2}$$
$$f_5 = 2r^2(x+a)^2 + p$$

IX. Si existe al menos un $T_\phi \epsilon Z$, $T_\emptyset \neq 0$ tal que $\phi \epsilon N$ entonces f_5 es un número compuesto, caso contrario f_5 es un número primo.

X. Si f_5 es un número compuesto con (T_ϕ, ϕ), $T_\phi \epsilon Z$, $\phi \epsilon Z$ y $\frac{-r \pm \phi}{2*T_\phi} = \frac{t}{b}$ con t y b primos entre sí entonces $2t^2 + pb^2$ o $\frac{2t^2+pb^2}{2}$ es un factor de f_5.

Algoritmo para f_6

$$\emptyset = \sqrt{-2pT_\emptyset^2 + 4M\,T_\emptyset + r_*^2}$$

Donde $r_* = 2rT - 1$ y $M = r(r_*(x+a) - 1)$

$$f_6 = pr_*^2 + 2M^2$$

XI. Si existe al menos un $T_\phi \epsilon Z$, $T_\emptyset \neq 0$ tal que $\phi \epsilon N$ entonces f_6 es un número compuesto, caso contrario f_6 es un número primo.

XII. Si f_6 es un número compuesto con (T_ϕ, ϕ), $T_\phi \epsilon Z$, $\phi \epsilon Z$ y $\frac{-r_* \pm \phi}{T_\phi} = \frac{t}{b}$ con t y b primos entre sí entonces $2t^2 + pb^2$ o $\frac{2t^2+pb^2}{2}$ es un factor de f_6.

FACTORIZACIÓN PRIMA DE LOS NÚMEROS POLINOMIALES DE LA FORMA $M^2R^2Q^2 + 2pr^2(M^2Q^2 + R^2Q^2 + M^2R^2)+4p^2r^4(M^2 + R^2 + Q^2) + 8p^3r^6$ CON $M = 2rn_1 + 1$, $R = 2rn_2 + 1$ y $Q = 2rn_3 + 1$

Sea $r, x, T, a \in Z$, $r \neq 0$ y $p \in \{3,5,11,29\}$.

Si:

$$n_1 = (4r^2x^2 + 4rx + 2pr^2 + 1) * T + 4rx^2 + 3x + 2pr$$

$$n_2 = (2r^2x^2 + p) * T + \frac{2r^2x^2 + 2r^2x + p - 1}{2r}$$

$$n_3 = (2r^2(x + a)^2 + p) * T - (\frac{2r^2(x + a)^2 + 2r^2(x + a) + p + 1}{2r})$$

Con $M = 2rn_1 + 1$, $R = 2rn_2 + 1$ y $Q = 2rn_3 + 1$

Entonces

$$M^2R^2Q^2 + 2pr^2(M^2Q^2 + R^2Q^2 + M^2R^2)+4p^2r^4(M^2 + R^2 + Q^2) + 8p^3r^6$$

es un número compuesto, con factores:

$$f_1 = 4r^2x^2 + 4rx + 2pr^2 + 1$$

$$f_2 = 16r^2(rT + 1)^2x^2 + 8r(2rT + 1)(rT + 1)x + 8r^2p(rT + 1)^2 + 4rT(rT + 1) + 1$$

$$f_3 = 2r^2x^2 + p$$

$$f_4 = 2r^2(2rT + 1)^2x^2 + 4r^2(2rT + 1)x + 4rpT(rT + 1) + 2r^2 + p$$

$$f_5 = 2r^2(x + a)^2 + p$$

$$f_6 = 2r^2[(2rT - 1)(x + a) - 1]^2 + p(2rT - 1)^2$$

Para encontrar la factorización prima del número polinomial $M^2R^2Q^2 + 2pr^2(M^2Q^2 + R^2Q^2 + M^2R^2)+4p^2r^4(M^2 + R^2 + Q^2) + 8p^3r^6$ debemos utilizar los seis algoritmos de Cordero para factorizar $f_1 - f_2 - f_3 - f_4 - f_5 - f_6$

Ejemplo de aplicación.

$sea\ r = 5, \qquad x = 8, \qquad T = 10, \qquad a = 7\ y\ \ p = 29$

$$n_1 = (4r^2x^2 + 4rx + 2pr^2 + 1) * T + 4rx^2 + 3x + 2pr$$

$$n_1 = 8011 * 10 + 1594 = 81704$$

$$n_2 = (2r^2x^2 + p) * T + \frac{2r^2x^2 + 2r^2x + p - 1}{2r}$$

$$n_2 = 3229 * 10 + \frac{1814}{5} = \frac{163264}{5}$$

$$n_3 = (2r^2(x + a)^2 + p) * T - \frac{2r^2(x + a)^2 + 2r^2(x + a) + p + 1}{2r}$$

$$n_3 = 11279 * 10 - 1203 = 111587$$

Calculemos el número polinomial que tenemos que factorizar en forma completa.

$$\boldsymbol{M = 2rn_1 + 1 = 2 * 5 * 81704 + 1 = 817041}$$

$$\boldsymbol{R = 2rn_2 + 1 = 2 * 5 * \frac{163264}{5} + 1 = 326529}$$

$$\boldsymbol{Q = 2rn_3 + 1 = 2 * 5 * 111587 + 1 = 1115871}$$

Tenemos que el número polinomial es:

$$\boldsymbol{M^2R^2Q^2 + 2pr^2(M^2Q^2 + R^2Q^2 + M^2R^2) + 4p^2r^4(M^2 + R^2 + Q^2) + 8p^3r^6}$$
$$= \boldsymbol{817041^2} * 326529^2 * \boldsymbol{1115871^2 + 58 * 25}$$
$$* (\boldsymbol{817041^2} * \boldsymbol{1115871^2} + 326529^2 * \boldsymbol{1115871^2} + \boldsymbol{817041^2}$$
$$* 326529^2) + \boldsymbol{4 * 29^2 * 5^4(817041^2 + 1115871^2} + 326529^2)$$
$$+ \boldsymbol{8 * 29^3 * 5^6}$$
$$= \boldsymbol{886256037593861437394368914559850 11}$$

Según el Teorema, tenemos que:

$$f_1 = 4r^2x^2 + 4rx + 2pr^2 + 1 = 8011$$

$$f_2 = 16r^2(rT + 1)^2x^2 + 8r(2rT + 1)(rT + 1)x + 8r^2p(rT + 1)^2$$
$$+ 4rT(rT + 1) + 1 = 83329921$$
$$f_3 = 2r^2x^2 + p = 3229$$

$$f_4 = 2r^2(2rT + 1)^2x^2 + 4r^2(2rT + 1)x + 4rpT(rT + 1) + 2r^2 + p$$

$$= 33019879$$

$$f_5 = 2r^2(x + a)^2 + p = 11279$$

$$f_6 = 2r^2[(2rT - 1)(x + a) - 1]^2 + p(2rT - 1)^2 = 110397029$$

Luego:

$$\boldsymbol{M^2R^2Q^2 + 2pr^2(M^2Q^2 + R^2Q^2 + M^2R^2) + 4p^2r^4(M^2 + R^2 + Q^2) + 8p^3r^6}$$
$$= \boldsymbol{817041^2} * 326529^2 * \boldsymbol{1115871^2 + 58 * 25}$$
$$* (\boldsymbol{817041^2 * 1115871^2} + 326529^2 * \boldsymbol{1115871^2 + 817041^2}$$
$$* 326529^2) + \boldsymbol{4 * 29^2 * 5^4(817041^2 + 1115871^2} + 326529^2)$$
$$+ \boldsymbol{8 * 29^3 * 5^6}$$
$$= \boldsymbol{886256037593861437394368914559850 11}$$
$$= \boldsymbol{8011 * 3229} * 83329921 * 11279 * 110397029 * 33019879$$

Pero generalmente esta no es la factorización prima, por lo que tenemos que utilizar los algoritmos de Cordero para encontrar su factorización prima.

Para

$$f_1 = 8011$$

Su algoritmo:

$$\emptyset = \sqrt{-2pT_\emptyset^2 + 2(2rx + 1)T_\emptyset + r^2}$$

$$\emptyset = \sqrt{-58T_\emptyset^2 + 162T_\emptyset + 25}$$

No encuentra pares ordenados enteros; por lo que:

$$f_1 = 8011$$

Es un número primo.

Para

$$f_2 = 83329921$$

Su algoritmo:

$$\emptyset = \sqrt{\frac{-1}{2}pT\emptyset^2 + \frac{1}{2}M\,T\emptyset + r_*^2}$$

$$Ø = \sqrt{\frac{-29}{2} T_Ø^{\ 2} + \frac{817041}{2} T_Ø + 65025}$$

Se encuentra solamente un par ordenado entero (14, 66), el cual nos indica que el número es biprimo y sus factores son:

$$\frac{t}{b} = \frac{r_* + Ø}{T_Ø} = \frac{255 + 66}{14} = \frac{321}{14}$$

Luego:

$$\frac{2t^2 + p * b^2}{2} = 105883$$

$$\frac{t}{b} = \frac{r_* - Ø}{T_Ø} = \frac{255 - 66}{14} = \frac{27}{2}$$

$$\frac{2t^2 + p * b^2}{2} = 787$$

La factorización completa de f_2 es:

$$f_2 = 83329921 = 787 * 105883$$

Para

$$f_3 = 3229$$

Su algoritmo:

$$Ø = \sqrt{-2pT_Ø^{\ 2} + 4r^2x\ T_Ø + r^2}$$

$$Ø = \sqrt{-58T_Ø^{\ 2} + 800\ T_Ø + 25}$$

No se encuentra pares ordenados enteros, lo cual nos indica que el número f_3 = 3229 es primo.

Para

$$f_4 = 2r^2(2rT + 1)^2x^2 + 4r^2(2rT + 1)x + 4rpT(rT + 1) + 2r^2 + p$$
$$= 33019879$$

Su algoritmo:

$$Ø = \sqrt{-2pT_Ø^2 + 4M\,T_Ø + r_*^2}$$

$$r_* = 2rT + 1 = 101 \text{ y } M = r(r_* x + 1) = 4045$$

$$f_4 = pr_*^2 + 2M^2 = 33019879$$

$$Ø = \sqrt{-58T^2 + 16180\,T}$$

Utilizando el algoritmo se obtiene el par ordenado entero (138, 1067), por lo que:

Luego:

$$\frac{t}{b} = \frac{-r_* + Ø}{2TØ} = \frac{101 + 1067}{2 * 138} = \frac{292}{69}$$

$$2 * 2922 + 29 * 692 = 308597$$

$$\frac{t}{b} = \frac{-r_* - Ø}{2TØ} = \frac{101 - 1067}{2 * 138} = \frac{-7}{2}$$

Luego:

$$\frac{2 * 7^2 + 29 * 2^2}{2} = 107$$

Así:

$$f_4 = 33019879 = 107 * 308597$$

Para

$$f_5 = 11279$$

Su algoritmo:

$$Ø = \sqrt{-2pT_Ø^2 + 4r^2(x + a)\,T_Ø + r^2}$$

$$Ø = \sqrt{-58T_Ø^2 + 1500\,T_Ø + 25}$$

No se encuentra pares ordenados enteros, lo cual nos indica que el número f_3 = 11279 es primo.

Ahora se puede factorizar completamente el número polinomial:

$$\mathbf{88625603759386143739436891455985011}$$
$$= 107 * 787 * 3229 * 8011 * 11279 * 105883 * 308597 * 110397029$$

Los Softwares o Programas Computacionales RRJ están basados en el algoritmo de Cordero r, n, p, el Segundo Teorema General de La Factorización de Cordero y en los algoritmos de los factores f_1 y f_2 para el Segundo Teorema General, f_3 y f_4 del Segundo Teorema General de La Factorización de Cordero, f_5 y f_6 del Segundo Teorema General de La Factorización de Cordero y $f_1 - f_2 - f_3 - f_4 - f_5 - f_6$ del Segundo Teorema General de La Factorización de Cordero.

Su nombre se deriva del nombre de sus coautores: El Costarricense Ronald Cordero Méndez, el Español Roberto Reinosa Fernández y el Costarricense Juan Francisco Gamboa Abarca.

CONCLUSIONES

1. Se comprueba la existencia de grupos infinitos de polinomios generadores de números primos que poseen las mismas características y que son útiles en la factorización de números enteros.
2. Los software comprueban la veracidad de los algoritmos y del teorema hasta números de cierta cantidad de dígitos.
3. Se necesita de tecnología mucho más potente para comprobar la veracidad y utilidad del teorema y de los algoritmos en la factorización de números enteros muy grandes.
4. La Investigación fomenta el interés por el estudio de los polinomios generadores de números primos, y su ya comprobada utilidad en la factorización de los números enteros.
5. Reconocemos que todavía falta mucho por caminar en el duro proceso de encontrar una solución definitiva en la factorización de los números enteros, pero la investigación contribuye a la búsqueda de tan esperada solución.

REFERENCIAS

Abel, U. y Siebert, H."Secuencias con un gran número de valores primos". Soy. Matemáticas. Mensual 100, 167-169, 1993.

Boston, N. y Greenwood, M. L. "Cuadráticas que representan números primos". América. Matemáticas. Mensual 102,595-599, 1995

Dudley, U. "Historia de la fórmula de los números primos". América. Matemáticas. Mensual 76, 23-28, 1969.

Garrison, B. “Polinomios con un gran número de valores primos”. América. Matemáticas. Mensual 97, 316-317, 1990.

Hardy, G. H. y Wright, E. M. “Introducción a la Teoría de Números”, 5° ed. Oxford, Inglaterra: Clarendon Press, 1979.

Pregg, E. Jr. “Concursos de programación de Al Zimmermann: polinomios generadores de primos”. 13 de marzo de 2006. https:// www.recmath.org/contest/description.php.