Códigos Cíclicos

Daniel Penazzi

June 15, 2021

Tabla de Contenidos

Códigos Cíclicos

- Una clase importante de códigos lineales son los códigos cíclicos.

Códigos Cíclicos

- Una clase importante de códigos lineales son los códigos cíclicos.
- Son una clase de códigos lineales con algunas propiedades extras que hacen que el codificado y decodificado de las palabras sea mas eficiente.

Códigos Cíclicos

- Una clase importante de códigos lineales son los códigos cíclicos.
- Son una clase de códigos lineales con algunas propiedades extras que hacen que el codificado y decodificado de las palabras sea mas eficiente.
- Ademas, en el caso de corrección de mas de un error, tienen un algoritmo mas eficiente que el que podria obtenerse para un código lineal general.

- Una clase importante de códigos lineales son los códigos cíclicos.
- Son una clase de códigos lineales con algunas propiedades extras que hacen que el codificado y decodificado de las palabras sea mas eficiente.
- Ademas, en el caso de corrección de mas de un error, tienen un algoritmo mas eficiente que el que podria obtenerse para un código lineal general.
- Y otra propiedad que tienen es que son muy eficientes para corregir “errores en ráfaga"

Códigos Cíclicos

- Una clase importante de códigos lineales son los códigos cíclicos.
- Son una clase de códigos lineales con algunas propiedades extras que hacen que el codificado y decodificado de las palabras sea mas eficiente.
- Ademas, en el caso de corrección de mas de un error, tienen un algoritmo mas eficiente que el que podria obtenerse para un código lineal general.
- Y otra propiedad que tienen es que son muy eficientes para corregir “errores en ráfaga"
- Es decir, cuando las condiciones son tales que es mas probable tener errores en un cierto segmento que aleatoriamente a lo largo de la transmisión.

Definición

Un código es cíclico si es lineal y la rotación de cualquiera de sus palabras es otra palabra del código.

Definición

Un código es cíclico si es lineal y la rotación de cualquiera de sus palabras es otra palabra del código.

- Por ejemplo $\{000, 011, 101, 110\}$ es cíclico pues cumple ambas propiedades.

Definición

Un código es cíclico si es lineal y la rotación de cualquiera de sus palabras es otra palabra del código.

- Por ejemplo $\{000, 011, 101, 110\}$ es cíclico pues cumple ambas propiedades.
- Pero $\{000, 001, 110, 111\}$ no lo es pues es lineal pero la rotación de la palabra 001 en un bit a izquierda o derecha (es decir, 010 o 100) no está en el código.

Códigos Cíclicos

Definición

Un código es cíclico si es lineal y la rotación de cualquiera de sus palabras es otra palabra del código.

- Por ejemplo $\{000, 011, 101, 110\}$ es cíclico pues cumple ambas propiedades.
- Pero $\{000, 001, 110, 111\}$ no lo es pues es lineal pero la rotación de la palabra 001 en un bit a izquierda o derecha (es decir, 010 o 100) no está en el código.
- $\{000, 001, 010, 100\}$ cumple que la rotación de cualquier palabra es una palabra del código, pero no es lineal, asi que no es un código cíclico.

Códigos Cíclicos

Definición

Un código es cíclico si es lineal y la rotación de cualquiera de sus palabras es otra palabra del código.

- Por ejemplo $\{000, 011, 101, 110\}$ es cíclico pues cumple ambas propiedades.
- Pero $\{000, 001, 110, 111\}$ no lo es pues es lineal pero la rotación de la palabra 001 en un bit a izquierda o derecha (es decir, 010 o 100) no está en el código.
- $\{000, 001, 010, 100\}$ cumple que la rotación de cualquier palabra es una palabra del código, pero no es lineal, asi que no es un código cíclico.
- A los códigos cíclicos también se los llama CRC (Cyclic redundancy code)

Cambio de notación

Notación

Por motivos que resultaran claros en breve, en vez de denotar las palabras como $w_1...w_n$, las denotaremos como $w_0...w_{n-1}$

Cambio de notación

Notación

Por motivos que resultaran claros en breve, en vez de denotar las palabras como $w_1 ... w_n$, las denotaremos como $w_0 ... w_{n-1}$

Definición

Dada una palabra $w = w_0 w_1 \ldots\ldots w_{n-2} w_{n-1}$, definimos la rotación (o cyclic shift) de w como la palabra $rot(w) = w_{n-1} w_0 w_1 \ldots\ldots w_{n-2}$

- Dado que $rot^2(w) = w_{n-2}w_{n-1}w_0...w_{n-3}$, etc, esta claro que C es cíclico si C es lineal y se cumple que $w \in C \Rightarrow rot(w) \in C$.

Códigos Cíclicos

- Dado que $rot^2(w) = w_{n-2}w_{n-1}w_0...w_{n-3}$, etc, esta claro que C es cíclico si C es lineal y se cumple que $w \in C \Rightarrow rot(w) \in C$.
- Como *rot* es lineal, tambien tenemos que un código es cíclico sii es lineal y existe una base de C tal que $rot(w) \in C$ para toda palabra w de la base.

Códigos Cíclicos

- Dado que $rot^2(w) = w_{n-2}w_{n-1}w_0...w_{n-3}$, etc, esta claro que C es cíclico si C es lineal y se cumple que $w \in C \Rightarrow rot(w) \in C$.
- Como *rot* es lineal, tambien tenemos que un código es cíclico sii es lineal y existe una base de C tal que $rot(w) \in C$ para toda palabra w de la base.
- Por lo tanto es fácil construir códigos cíclicos:

Códigos Cíclicos

- Dado que $rot^2(w) = w_{n-2}w_{n-1}w_0...w_{n-3}$, etc, esta claro que C es cíclico si C es lineal y se cumple que $w \in C \Rightarrow rot(w) \in C$.
- Como *rot* es lineal, tambien tenemos que un código es cíclico sii es lineal y existe una base de C tal que $rot(w) \in C$ para toda palabra w de la base.
- Por lo tanto es fácil construir códigos cíclicos:
  - Basta tomar una palabra w cualquiera, y tomar el espacio vectorial generado por el conjunto $\{w, rot(w), rot^2(w), ..., rot^{n-1}(w)\}$.

- Dado que $rot^2(w) = w_{n-2}w_{n-1}w_0...w_{n-3}$, etc, esta claro que C es cíclico si C es lineal y se cumple que $w \in C \Rightarrow rot(w) \in C$.
- Como *rot* es lineal, tambien tenemos que un código es cíclico sii es lineal y existe una base de C tal que $rot(w) \in C$ para toda palabra w de la base.
- Por lo tanto es fácil construir códigos cíclicos:
  - Basta tomar una palabra w cualquiera, y tomar el espacio vectorial generado por el conjunto $\{w, rot(w), rot^2(w), ..., rot^{n-1}(w)\}$.
- Como $rot^n(w) = w$, ese conjunto es cerrado por la operacion *rot*

Códigos Cíclicos

- Dado que $rot^2(w) = w_{n-2}w_{n-1}w_0...w_{n-3}$, etc, esta claro que C es cíclico si C es lineal y se cumple que $w \in C \Rightarrow rot(w) \in C$.
- Como *rot* es lineal, tambien tenemos que un código es cíclico sii es lineal y existe una base de C tal que $rot(w) \in C$ para toda palabra w de la base.
- Por lo tanto es fácil construir códigos cíclicos:
  - Basta tomar una palabra w cualquiera, y tomar el espacio vectorial generado por el conjunto $\{w, rot(w), rot^2(w), ..., rot^{n-1}(w)\}$.
- Como $rot^n(w) = w$, ese conjunto es cerrado por la operacion *rot*
- Como genera C, podemos extraer una base del mismo que cumple la propiedad anterior.

- El problema con hacer esto es que no tenemos la menor idea de cual sera la dimensión de C, por ejemplo.

- El problema con hacer esto es que no tenemos la menor idea de cual sera la dimensión de C, por ejemplo.
- Veremos que podemos elejir la palabra w mas cuidadosamente, de forma tal de obtener una base que consistirá en las primeras k rotaciónes de w.

Códigos Cíclicos

- El problema con hacer esto es que no tenemos la menor idea de cual sera la dimensión de C, por ejemplo.
- Veremos que podemos elejir la palabra w mas cuidadosamente, de forma tal de obtener una base que consistirá en las primeras k rotaciónes de w.
- Ademas, esta palabra w especial tendra otras propiedades que la harán muy efectiva.

Códigos Cíclicos

- El problema con hacer esto es que no tenemos la menor idea de cual sera la dimensión de C, por ejemplo.
- Veremos que podemos elejir la palabra w mas cuidadosamente, de forma tal de obtener una base que consistirá en las primeras k rotaciónes de w.
- Ademas, esta palabra w especial tendra otras propiedades que la harán muy efectiva.
- En particular, en vez de tener que guardar toda una matriz $k \times n$ generadora, o $r \times (r + k)$ de chequeo, bastará guardar una sola palabra de longitud n.

Álgebra

- Hay una cosa que debería quedar claro en el secundario, pero nunca o casi nunca la enseñan bien.

Álgebra

- Hay una cosa que debería quedar claro en el secundario, pero nunca o casi nunca la enseñan bien.
- Cuando yo era estudiante, nos enseñaban bien esta diferencia en primer año de famaf, pero creo que la calidad ha decaido y no estoy seguro si siguen haciendolo.

Álgebra

- Hay una cosa que debería quedar claro en el secundario, pero nunca o casi nunca la enseñan bien.
- Cuando yo era estudiante, nos enseñaban bien esta diferencia en primer año de famaf, pero creo que la calidad ha decaido y no estoy seguro si siguen haciendolo.
- Asi que repasaremos un poco acerca de polinomios.

Álgebra

- Hay una cosa que debería quedar claro en el secundario, pero nunca o casi nunca la enseñan bien.
- Cuando yo era estudiante, nos enseñaban bien esta diferencia en primer año de famaf, pero creo que la calidad ha decaido y no estoy seguro si siguen haciendolo.
- Asi que repasaremos un poco acerca de polinomios.
- Todos “sabemos" que los polinomios son “cosas" como $1+x, 2+x^2, x+x^4+5x^7+x^{10}$, etc

Álgebra

- Hay una cosa que debería quedar claro en el secundario, pero nunca o casi nunca la enseñan bien.
- Cuando yo era estudiante, nos enseñaban bien esta diferencia en primer año de famaf, pero creo que la calidad ha decaido y no estoy seguro si siguen haciendolo.
- Asi que repasaremos un poco acerca de polinomios.
- Todos “sabemos" que los polinomios son “cosas" como $1 + x, 2 + x^2, x + x^4 + 5x^7 + x^{10}$, etc
- En general, algo de la forma $\sum_{i=0}^{d} a_i x^i$.

Álgebra

- Hay una cosa que debería quedar claro en el secundario, pero nunca o casi nunca la enseñan bien.
- Cuando yo era estudiante, nos enseñaban bien esta diferencia en primer año de famaf, pero creo que la calidad ha decaido y no estoy seguro si siguen haciendolo.
- Asi que repasaremos un poco acerca de polinomios.
- Todos “sabemos" que los polinomios son “cosas" como $1 + x, 2 + x^2, x + x^4 + 5x^7 + x^{10}$, etc
- En general, algo de la forma $\sum_{i=0}^{d} a_i x^i$.
- Pero ¿qué son, exactamente, esas “cosas"?

Polinomios

- Uno esta tentado a decir que son funciones, pero eso esta mal.

Polinomios

- Uno esta tentado a decir que son funciones, pero eso esta mal.
- Una función polinómica es una función de la forma $f(x) = \sum_{i=0}^{d} a_i x^i$, definida en algún lugar donde tenga sentido la suma y el producto con algunas propiedades minimas, es decir, en un anillo.

Polinomios

- Uno esta tentado a decir que son funciones, pero eso esta mal.
- Una función polinómica es una función de la forma $f(x) = \sum_{i=0}^{d} a_i x^i$, definida en algún lugar donde tenga sentido la suma y el producto con algunas propiedades minimas, es decir, en un anillo.
- Pero eso no es un polinomio.

Polinomios

- Uno esta tentado a decir que son funciones, pero eso esta mal.
- Una función polinómica es una función de la forma $f(x) = \sum_{i=0}^{d} a_i x^i$, definida en algún lugar donde tenga sentido la suma y el producto con algunas propiedades minimas, es decir, en un anillo.
- Pero eso no es un polinomio.
- La confusión viene porque en $\mathbb{R}$ las dos cosas se pueden identificar: toda función polinómica "es" un polinomio y viceversa.

Polinomios

- Uno esta tentado a decir que son funciones, pero eso esta mal.
- Una función polinómica es una función de la forma $f(x) = \sum_{i=0}^{d} a_i x^i$, definida en algún lugar donde tenga sentido la suma y el producto con algunas propiedades minimas, es decir, en un anillo.
- Pero eso no es un polinomio.
- La confusión viene porque en $\mathbb{R}$ las dos cosas se pueden identificar: toda función polinómica "es" un polinomio y viceversa.
- Pero en otros anillos eso no pasa.

Polinomios

- La propiedad fundamental que tienen los polinomios es que si $\sum_{i=0}^{d} a_i x^i = \sum_{i=0}^{r} b_i x^i$ con a_d, b_r no nulos, entonces $d = r$ y $a_i = b_i$ para todo i.

Polinomios

- La propiedad fundamental que tienen los polinomios es que si $\sum_{i=0}^{d} a_i x^i = \sum_{i=0}^{r} b_i x^i$ con a_d, b_r no nulos, entonces $d = r$ y $a_i = b_i$ para todo i.
- Entonces por ejemplo en $\{0, 1\}$, los polinomios $1 + x$ y $1 + x^2$ son distintos, pero las funciones polinomicas $x \mapsto 1 + x$ y $1 + x^2$ son iguales, pues dos funciones f, g son iguales sii $f(x) = g(x)$ para todo x, y $1 + x = 1 + x^2$ para todo $x \in \{0, 1\}$.

Polinomios

- La propiedad fundamental que tienen los polinomios es que si $\sum_{i=0}^{d} a_i x^i = \sum_{i=0}^{r} b_i x^i$ con a_d, b_r no nulos, entonces $d = r$ y $a_i = b_i$ para todo i.
- Entonces por ejemplo en $\{0, 1\}$, los polinomios $1 + x$ y $1 + x^2$ son distintos, pero las funciones polinomicas $x \mapsto 1 + x$ y $1 + x^2$ son iguales, pues dos funciones f, g son iguales sii $f(x) = g(x)$ para todo x, y $1 + x = 1 + x^2$ para todo $x \in \{0, 1\}$.
- Asi que en general se define un polinomio simplemente como la “suma formal" $\sum_{i=0}^{d} a_i x^i$.

Polinomios

- La propiedad fundamental que tienen los polinomios es que si $\sum_{i=0}^{d} a_i x^i = \sum_{i=0}^{r} b_i x^i$ con a_d, b_r no nulos, entonces $d = r$ y $a_i = b_i$ para todo i.
- Entonces por ejemplo en $\{0, 1\}$, los polinomios $1 + x$ y $1 + x^2$ son distintos, pero las funciones polinomicas $x \mapsto 1 + x$ y $1 + x^2$ son iguales, pues dos funciones f, g son iguales sii $f(x) = g(x)$ para todo x, y $1 + x = 1 + x^2$ para todo $x \in \{0, 1\}$.
- Asi que en general se define un polinomio simplemente como la “suma formal" $\sum_{i=0}^{d} a_i x^i$.
- Esto parece un acto de magia, pero se puede definir formalmente.

Polinomios

- La propiedad fundamental que tienen los polinomios es que si $\sum_{i=0}^{d} a_i x^i = \sum_{i=0}^{r} b_i x^i$ con a_d, b_r no nulos, entonces $d = r$ y $a_i = b_i$ para todo i.
- Entonces por ejemplo en $\{0, 1\}$, los polinomios $1 + x$ y $1 + x^2$ son distintos, pero las funciones polinomicas $x \mapsto 1 + x$ y $1 + x^2$ son iguales, pues dos funciones f, g son iguales sii $f(x) = g(x)$ para todo x, y $1 + x = 1 + x^2$ para todo $x \in \{0, 1\}$.
- Asi que en general se define un polinomio simplemente como la "suma formal" $\sum_{i=0}^{d} a_i x^i$.
- Esto parece un acto de magia, pero se puede definir formalmente.
- Hay varias formas, ahora explico una que es la que nos será útil.

Polinomios

- Se define “x" como la palabra infinita 010.....

Polinomios

- Se define "x" como la palabra infinita 010.....
- x^2 como 001000...

Polinomios

- Se define “x" como la palabra infinita 010.....
- x^2 como 001000...
- y en general x^i como la palabra infinita a derecha que tiene un 1 en la posición im contando desde 0, y cero en las otras.

Polinomios

- Se define “x" como la palabra infinita 010.....
- x^2 como 001000...
- y en general x^i como la palabra infinita a derecha que tiene un 1 en la posición im contando desde 0, y cero en las otras.
- Y la suma y multiplicación de constantes de la forma obvia.

Polinomios

- Se define “x" como la palabra infinita 010.....
- x^2 como 001000...
- y en general x^i como la palabra infinita a derecha que tiene un 1 en la posición *i*m contando desde 0, y cero en las otras.
- Y la suma y multiplicación de constantes de la forma obvia.
- Asi, $1 + x^4 + 5x^7 + 8x^{10} = 10001005008 00....$

Polinomios

- Se define "x" como la palabra infinita 010.....
- x^2 como 001000...
- y en general x^i como la palabra infinita a derecha que tiene un 1 en la posición im contando desde 0, y cero en las otras.
- Y la suma y multiplicación de constantes de la forma obvia.
- Asi, $1 + x^4 + 5x^7 + 8x^{10} = 1000100500800....$
- Un polinomio entonces será simplemente una palabra infinita pero tal que tenga una cantidad finita de entradas no nulas.

Polinomios

- Se define "x" como la palabra infinita 010.....
- x^2 como 001000...
- y en general x^i como la palabra infinita a derecha que tiene un 1 en la posición im contando desde 0, y cero en las otras.
- Y la suma y multiplicación de constantes de la forma obvia.
- Asi, $1 + x^4 + 5x^7 + 8x^{10} = 1000100500800....$
- Un polinomio entonces será simplemente una palabra infinita pero tal que tenga una cantidad finita de entradas no nulas.
- Asi que las entradas infinitas nulas a derecha pueden no escribirse y se puede escribir $1 + x^4 + 5x^7 + 8x^{10} = 10001005008$ entendiendo que luego siguen todos ceros.

Polinomios

- Se define "x" como la palabra infinita 010.....
- x^2 como 001000...
- y en general x^i como la palabra infinita a derecha que tiene un 1 en la posición *i*m contando desde 0, y cero en las otras.
- Y la suma y multiplicación de constantes de la forma obvia.
- Asi, $1 + x^4 + 5x^7 + 8x^{10} = 1000100500800....$
- Un polinomio entonces será simplemente una palabra infinita pero tal que tenga una cantidad finita de entradas no nulas.
- Asi que las entradas infinitas nulas a derecha pueden no escribirse y se puede escribir $1 + x^4 + 5x^7 + 8x^{10} = 10001005008$ entendiendo que luego siguen todos ceros.
- Luego se define la multiplicación entre polinomios de forma tal que obedezca las reglas que ya conocemos.

Lo importante

- Si no leyeron nada de lo anterior, o leyeron pero no entendieron, lo importante que les debe quedar, y que es lo que vamos a usar es lo siguiente:

Lo importante

- Si no leyeron nada de lo anterior, o leyeron pero no entendieron, lo importante que les debe quedar, y que es lo que vamos a usar es lo siguiente:

Clave

La palabra $w_0 w_1 .... w_{n-1}$ se puede pensar como el polinomio $w_0 + w_1 x + w_2 x^2 + \cdots + w_{n-1} x^{n-1}$.

Lo importante

- Si no leyeron nada de lo anterior, o leyeron pero no entendieron, lo importante que les debe quedar, y que es lo que vamos a usar es lo siguiente:

Clave

La palabra $w_0 w_1 ....w_{n-1}$ se puede pensar como el polinomio $w_0 + w_1 x + w_2 x^2 + \cdots + w_{n-1} x^{n-1}$.

- Por ejemplo, $1010 = 1 + x^2$

Lo importante

- Si no leyeron nada de lo anterior, o leyeron pero no entendieron, lo importante que les debe quedar, y que es lo que vamos a usar es lo siguiente:

Clave

La palabra $w_0 w_1 .... w_{n-1}$ se puede pensar como el polinomio $w_0 + w_1 x + w_2 x^2 + \cdots + w_{n-1} x^{n-1}$.

- Por ejemplo, $1010 = 1 + x^2$
- Advertencia: en algunos textos la identificación es asumiendo que el termino de mas a la izquierda es el termino de MAYOR grado. En ese caso, 1010 representa al polinomio $x^3 + x$ y no al $1 + x^2$, asi que hay que prestar atención a cual identificación se hace.

Palabras y polinomios

- ¿Que ganamos pensando en una palabra como un polinomio?

Palabras y polinomios

- ¿Que ganamos pensando en una palabra como un polinomio?
- Que los polinomios se pueden multiplicar.

Palabras y polinomios

- ¿Que ganamos pensando en una palabra como un polinomio?
- Que los polinomios se pueden multiplicar.
- Y entonces se pueden mirar los códigos con una estructura algebraica mas “rica" que permite deducir propiedades.

Palabras y polinomios

- ¿Que ganamos pensando en una palabra como un polinomio?
- Que los polinomios se pueden multiplicar.
- Y entonces se pueden mirar los códigos con una estructura algebraica mas “rica" que permite deducir propiedades.
- Pero hay un small problem.

Palabras y polinomios

- ¿Que ganamos pensando en una palabra como un polinomio?
- Que los polinomios se pueden multiplicar.
- Y entonces se pueden mirar los códigos con una estructura algebraica mas “rica" que permite deducir propiedades.
- Pero hay un small problem.
- Dado que estamos trabajando con códigos de longitud n, entonces estamos trabajando con polinomios de grado menor que n. (es decir, el termino no nulo de grado mas alto es $n-1$ o menor)

Palabras y polinomios

- ¿Que ganamos pensando en una palabra como un polinomio?
- Que los polinomios se pueden multiplicar.
- Y entonces se pueden mirar los códigos con una estructura algebraica mas “rica" que permite deducir propiedades.
- Pero hay un small problem.
- Dado que estamos trabajando con códigos de longitud n, entonces estamos trabajando con polinomios de grado menor que n. (es decir, el termino no nulo de grado mas alto es $n-1$ o menor)
- Y si multiplicamos polinomios el grado crece.

- Por ejemplo, si multiplicaramos:

Palabras y polinomios

- Por ejemplo, si multiplicaramos:
- $1010.0110 = (1 + x^2)(x + x^2) = x + x^2 + x^3 + x^4 = 01111$

Palabras y polinomios

- Por ejemplo, si multiplicaramos:
- $1010.0110 = (1 + x^2)(x + x^2) = x + x^2 + x^3 + x^4 = 01111$
- Pasariamos de palabras de longitud 4 a palabras de longitud 5.

Palabras y polinomios

- Por ejemplo, si multiplicaramos:
- $1010.0110 = (1 + x^2)(x + x^2) = x + x^2 + x^3 + x^4 = 01111$
- Pasariamos de palabras de longitud 4 a palabras de longitud 5.
- Pero queremos quedarnos “dentro" de las palabras de longitud 4, pues queremos trabajar con códigos de bloque.

Palabras y polinomios

- Por ejemplo, si multiplicaramos:
- $1010.0110 = (1 + x^2)(x + x^2) = x + x^2 + x^3 + x^4 = 01111$
- Pasariamos de palabras de longitud 4 a palabras de longitud 5.
- Pero queremos quedarnos “dentro" de las palabras de longitud 4, pues queremos trabajar con códigos de bloque.
- Para resolver ese problema, tomamos módulo, porque otra cosa que se puede hacer con polinomios es dividir.

Códigos Cíclicos

Definición

Si $p(x)$ y $m(x)$ son polinomios, entonces "$p(x) \bmod m(x)$" denotará el resto de la division de $p(x)$ por $m(x)$.
Es decir, $p(x) \bmod m(x)$ es el único polinomio $r(x)$ de grado menor que el grado de $m(x)$ tal que existe un polimonio $q(x)$ con $p(x) = q(x)m(x) + r(x)$

Códigos Cíclicos

Definición

Si $p(x)$ y $m(x)$ son polinomios, entonces "$p(x) \bmod m(x)$" denotará el resto de la division de $p(x)$ por $m(x)$.
Es decir, $p(x) \bmod m(x)$ es el único polinomio $r(x)$ de grado menor que el grado de $m(x)$ tal que existe un polimonio $q(x)$ con $p(x) = q(x)m(x) + r(x)$

- Tambien diremos que $p(x) \equiv q(x)_{(\bmod h(x))}$ sii:

Definición

Si $p(x)$ y $m(x)$ son polinomios, entonces "$p(x) \mod m(x)$" denotará el resto de la division de $p(x)$ por $m(x)$.
Es decir, $p(x) \mod m(x)$ es el único polinomio $r(x)$ de grado menor que el grado de $m(x)$ tal que existe un polimonio $q(x)$ con $p(x) = q(x)m(x) + r(x)$

- Tambien diremos que $p(x) \equiv q(x)_{(\mod h(x))}$ sii:
  - $p(x) \mod h(x) = q(x) \mod h(x)$.

- Por lo tanto, si queremos “multiplicar" dos palabras de longitud n y obtener otra vez una palabra de longitud n

- Por lo tanto, si queremos “multiplicar" dos palabras de longitud n y obtener otra vez una palabra de longitud n
  - es decir, multiplicar dos polinomios de grado menor que n y obtener otro polinomio de grado menor que n

- Por lo tanto, si queremos “multiplicar" dos palabras de longitud n y obtener otra vez una palabra de longitud n
  - es decir, multiplicar dos polinomios de grado menor que n y obtener otro polinomio de grado menor que n
- bastará con multiplicar los polinomios correspondientes y luego tomar modulo algun polinomio de grado n.

- Por lo tanto, si queremos “multiplicar" dos palabras de longitud n y obtener otra vez una palabra de longitud n
  - es decir, multiplicar dos polinomios de grado menor que n y obtener otro polinomio de grado menor que n
- bastará con multiplicar los polinomios correspondientes y luego tomar modulo algun polinomio de grado n.
- Por ejemplo, podriamos tomar el producto módulo x^n.

- Por lo tanto, si queremos “multiplicar" dos palabras de longitud n y obtener otra vez una palabra de longitud n
  - es decir, multiplicar dos polinomios de grado menor que n y obtener otro polinomio de grado menor que n
- bastará con multiplicar los polinomios correspondientes y luego tomar modulo algun polinomio de grado n.
- Por ejemplo, podriamos tomar el producto módulo x^n.
- Pero será mejor tomar el producto módulo $1 + x^n$

- Por lo tanto, si queremos “multiplicar" dos palabras de longitud n y obtener otra vez una palabra de longitud n
  - es decir, multiplicar dos polinomios de grado menor que n y obtener otro polinomio de grado menor que n
- bastará con multiplicar los polinomios correspondientes y luego tomar modulo algun polinomio de grado n.
- Por ejemplo, podriamos tomar el producto módulo x^n.
- Pero será mejor tomar el producto módulo $1 + x^n$
- Para no confundirnos con la multiplicación usual de polinomios, la denotaremos con un simbolo especial

Códigos Cíclicos

Notación

Dadas dos palabras v y w de longitud n, identificadas con los polinomios $v(x)$, $w(x)$, definimos:

$$v \odot w = v(x)w(x) \mod (1 + x^n)$$

Notación

Dadas dos palabras v y w de longitud n, identificadas con los polinomios $v(x)$, $w(x)$, definimos:

$$v \odot w = v(x)w(x) \mod (1 + x^n)$$

- Nota: en ocasiones extenderemos la definición a casos donde una de las palabras tenga mas de n bits, definiendola de la misma forma.

Códigos Cíclicos

Notación

Dadas dos palabras v y w de longitud n, identificadas con los polinomios $v(x)$, $w(x)$, definimos:

$$v \odot w = v(x)w(x) \mod (1 + x^n)$$

- Nota: en ocasiones extenderemos la definición a casos donde una de las palabras tenga mas de n bits, definiendola de la misma forma.
- Ejemplo: Si $n = 4$ tenemos:

$$1010 \odot 0110 \quad = \quad (1 + x^2)(x + x^2) \mod 1 + x^4$$

Códigos Cíclicos

Notación

Dadas dos palabras v y w de longitud n, identificadas con los polinomios $v(x)$, $w(x)$, definimos:

$$v \odot w = v(x)w(x) \mod (1 + x^n)$$

- Nota: en ocasiones extenderemos la definición a casos donde una de las palabras tenga mas de n bits, definiendola de la misma forma.
- Ejemplo: Si $n = 4$ tenemos:

$$\begin{aligned} 1010 \odot 0110 &= (1 + x^2)(x + x^2) \bmod 1 + x^4 \\ &= (x + x^2 + x^3 + x^4) \bmod 1 + x^4 \end{aligned}$$

Códigos Cíclicos

Notación

Dadas dos palabras v y w de longitud n, identificadas con los polinomios $v(x)$, $w(x)$, definimos:

$$v \odot w = v(x)w(x) \mod (1 + x^n)$$

- Nota: en ocasiones extenderemos la definición a casos donde una de las palabras tenga mas de n bits, definiendola de la misma forma.
- Ejemplo: Si $n = 4$ tenemos:

$$\begin{aligned} 1010 \odot 0110 &= (1 + x^2)(x + x^2) \bmod 1 + x^4 \\ &= (x + x^2 + x^3 + x^4) \bmod 1 + x^4 \\ &= 1 + x + x^2 + x^3 = 1111 \end{aligned}$$

Códigos Cíclicos

- Para tomar módulo $1 + x^n$ no hace falta dividir el polinomio por $1 + x^n$ y calcular el resto.

- Para tomar módulo $1+x^n$ no hace falta dividir el polinomio por $1+x^n$ y calcular el resto.
- Basta recordar que $\mod$ es lineal

- Para tomar módulo $1 + x^n$ no hace falta dividir el polinomio por $1 + x^n$ y calcular el resto.
- Basta recordar que $\mod$ es lineal
- Por lo tanto,como $(1 + x^n) \mod (1 + x^n) = 0$, entonces obviamente $x^n \mod (1 + x^n) = 1$.

- Para tomar módulo $1+x^n$ no hace falta dividir el polinomio por $1+x^n$ y calcular el resto.
- Basta recordar que $\mod$ es lineal
- Por lo tanto,como $(1+x^n) \mod (1+x^n)=0$, entonces obviamente $x^n \mod (1+x^n)=1$.
- (esto ultimo pues estamos trabajando en $\{0,1\}$)

Códigos Cíclicos

- Para tomar módulo $1 + x^n$ no hace falta dividir el polinomio por $1 + x^n$ y calcular el resto.
- Basta recordar que $\bmod$ es lineal
- Por lo tanto,como $(1 + x^n) \bmod (1 + x^n) = 0$, entonces obviamente $x^n \bmod (1 + x^n) = 1$.
- (esto ultimo pues estamos trabajando en $\{0, 1\}$)
- Se puede ver esto directamente: $x^n = (1 + x^n).1 + 1$.

- Para tomar módulo $1 + x^n$ no hace falta dividir el polinomio por $1 + x^n$ y calcular el resto.
- Basta recordar que $\mod$ es lineal
- Por lo tanto,como $(1 + x^n) \mod (1 + x^n) = 0$, entonces obviamente $x^n \mod (1 + x^n) = 1$.
- (esto ultimo pues estamos trabajando en $\{0, 1\}$)
- Se puede ver esto directamente: $x^n = (1 + x^n).1 + 1$.
- En general si se trabaja en entornos donde $1 \neq -1$, en vez de tomar el polinomio $1 + x^n$ como módulo, se toma el polinomio $-1 + x^n$.

Códigos Cíclicos

- Para tomar módulo $1 + x^n$ no hace falta dividir el polinomio por $1 + x^n$ y calcular el resto.
- Basta recordar que $\mod$ es lineal
- Por lo tanto,como $(1 + x^n) \mod (1 + x^n) = 0$, entonces obviamente $x^n \mod (1 + x^n) = 1$.
- (esto ultimo pues estamos trabajando en $\{0, 1\}$)
- Se puede ver esto directamente: $x^n = (1 + x^n).1 + 1$.
- En general si se trabaja en entornos donde $1 \neq -1$, en vez de tomar el polinomio $1 + x^n$ como módulo, se toma el polinomio $-1 + x^n$.
- Pues $x^n \mod (-1 + x^n) = 1$

Clave para la utilidad de los códigos cíclicos

Propiedad

$$rot(w) = x \odot w(x)$$

Clave para la utilidad de los códigos cíclicos

Propiedad

$$rot(w) = x \odot w(x)$$

Prueba:

Clave para la utilidad de los códigos cíclicos

Propiedad

$$rot(w) = x \odot w(x)$$

Prueba:

$$x \odot w(x) \quad = \quad x(w_0 + w_1 x + \ldots + w_{n-2}x^{n-2} + w_{n-1}x^{n-1}) \bmod (1 + x^n)$$

Clave para la utilidad de los códigos cíclicos

Propiedad

$$rot(w) = x \odot w(x)$$

Prueba:

$$\begin{aligned} x \odot w(x) &= x(w_0 + w_1 x + \ldots + w_{n-2}x^{n-2} + w_{n-1}x^{n-1}) \bmod (1 + x^n) \\ &= (w_0 x + w_1 x^2 + \ldots + w_{n-2}x^{n-1} + w_{n-1}x^n) \bmod (1 + x^n) \end{aligned}$$

Clave para la utilidad de los códigos cíclicos

Propiedad

$$rot(w) = x \odot w(x)$$

Prueba:

$$\begin{aligned} x \odot w(x) &= x(w_0 + w_1 x + \ldots + w_{n-2}x^{n-2} + w_{n-1}x^{n-1}) \bmod (1 + x^n) \\ &= (w_0 x + w_1 x^2 + \ldots + w_{n-2}x^{n-1} + w_{n-1}x^n) \bmod (1 + x^n) \\ &= w_0 x + w_1 x^2 + \ldots + w_{n-2}x^{n-1} + w_{n-1} \end{aligned}$$

Clave para la utilidad de los códigos cíclicos

Propiedad

$$rot(w) = x \odot w(x)$$

Prueba:

$$\begin{aligned}
x \odot w(x) &= x(w_0 + w_1 x + ... + w_{n-2}x^{n-2} + w_{n-1}x^{n-1}) \bmod (1 + x^n) \\
&= (w_0 x + w_1 x^2 + ... + w_{n-2}x^{n-1} + w_{n-1}x^n) \bmod (1 + x^n) \\
&= w_0 x + w_1 x^2 + ... + w_{n-2}x^{n-1} + w_{n-1} \\
&= w_{n-1} + w_0 x + w_1 x^2 + ... + w_{n-2}x^{n-1}
\end{aligned}$$

Clave para la utilidad de los códigos cíclicos

Propiedad

$$rot(w) = x \odot w(x)$$

Prueba:

$$\begin{aligned}
x \odot w(x) &= x(w_0 + w_1 x + \ldots + w_{n-2}x^{n-2} + w_{n-1}x^{n-1}) \bmod (1 + x^n) \\
&= (w_0 x + w_1 x^2 + \ldots + w_{n-2}x^{n-1} + w_{n-1}x^n) \bmod (1 + x^n) \\
&= w_0 x + w_1 x^2 + \ldots + w_{n-2}x^{n-1} + w_{n-1} \\
&= w_{n-1} + w_0 x + w_1 x^2 + \ldots + w_{n-2}x^{n-1} \\
&= rot(w)
\end{aligned}$$

Clave para la utilidad de los códigos cíclicos

Propiedad

Sea C un código cíclico, $w \in C$ y v una palabra cualquiera. Entonces $v \odot w \in C$.

Clave para la utilidad de los códigos cíclicos

Propiedad

Sea C un código cíclico, $w \in C$ y v una palabra cualquiera. Entonces $v \odot w \in C$.

- Prueba: por la propiedad anterior, $x \odot w = rot(w) \in C$ (pues C es cíclico)

Clave para la utilidad de los códigos cíclicos

Propiedad

Sea C un código cíclico, $w \in C$ y v una palabra cualquiera. Entonces $v \odot w \in C$.

- Prueba: por la propiedad anterior, $x \odot w = rot(w) \in C$ (pues C es cíclico)
- Por lo tanto $x^i \odot w \in C$ para todo i.

Clave para la utilidad de los códigos cíclicos

Propiedad

Sea C un código cíclico, $w \in C$ y v una palabra cualquiera. Entonces $v \odot w \in C$.

- Prueba: por la propiedad anterior, $x \odot w = rot(w) \in C$ (pues C es cíclico)
- Por lo tanto $x^i \odot w \in C$ para todo i.
- Como C, al ser cíclico, es lineal, entonces cualquier combinación lineal de $x^i \odot w$ estará en C.

Clave para la utilidad de los códigos cíclicos

Propiedad

Sea C un código cíclico, $w \in C$ y v una palabra cualquiera. Entonces $v \odot w \in C$.

- Prueba: por la propiedad anterior, $x \odot w = rot(w) \in C$ (pues C es cíclico)
- Por lo tanto $x^i \odot w \in C$ para todo i.
- Como C, al ser cíclico, es lineal, entonces cualquier combinación lineal de $x^i \odot w$ estará en C.
- Es decir, $\sum a_i(x^i \odot w) \in C$ para cualesquiera a_i.

Clave para la utilidad de los códigos cíclicos

Propiedad

Sea C un código cíclico, $w \in C$ y v una palabra cualquiera. Entonces $v \odot w \in C$.

- Prueba: por la propiedad anterior, $x \odot w = rot(w) \in C$ (pues C es cíclico)
- Por lo tanto $x^i \odot w \in C$ para todo i.
- Como C, al ser cíclico, es lineal, entonces cualquier combinación lineal de $x^i \odot w$ estará en C.
- Es decir, $\sum a_i(x^i \odot w) \in C$ para cualesquiera a_i.
- Pero $\sum a_i(x^i \odot w) = (\sum a_i x^i) \odot w$.

Clave para la utilidad de los códigos cíclicos

Propiedad

Sea C un código cíclico, $w \in C$ y v una palabra cualquiera. Entonces $v \odot w \in C$.

- Prueba: por la propiedad anterior, $x \odot w = rot(w) \in C$ (pues C es cíclico)
- Por lo tanto $x^i \odot w \in C$ para todo i.
- Como C, al ser cíclico, es lineal, entonces cualquier combinación lineal de $x^i \odot w$ estará en C.
- Es decir, $\sum a_i(x^i \odot w) \in C$ para cualesquiera a_i.
- Pero $\sum a_i(x^i \odot w) = (\sum a_i x^i) \odot w$.
- Concluimos que $v \odot w \in C$ para cualesquiera $v = \sum a_i x^i$.

Ideales

- En matemática a un objeto que tiene esa propiedad “absorbente" se le llama un ideal.

Ideales

- En matemática a un objeto que tiene esa propiedad “absorbente" se le llama un ideal.
- Asi que un código cíclico es un ideal.

Ideales

- En matemática a un objeto que tiene esa propiedad “absorbente" se le llama un ideal.
- Asi que un código cíclico es un ideal.
- Para seguir con las propiedades de códigos cíclicos, enunciaremos una propiedad que vale para cualquier código lineal.

Ideales

- En matemática a un objeto que tiene esa propiedad “absorbente" se le llama un ideal.
- Asi que un código cíclico es un ideal.
- Para seguir con las propiedades de códigos cíclicos, enunciaremos una propiedad que vale para cualquier código lineal.
- Sólo que es una propiedad útil exclusivamente en el caso de los códigos cíclicos, por eso no la dimos antes.

Propiedad

Si C es lineal, entonces existe **un único** polinomio no nulo en C de grado mínimo

Propiedad

Si C es lineal, entonces existe **un único** polinomio no nulo en C de grado mínimo

- Nota: esta propiedad vale sólo en $\{0, 1\}$. Si no estamos en $\{0, 1\}$ hay que agregar la condición de que sea mónico para la unicidad.

Propiedad

Si C es lineal, entonces existe **un único** polinomio no nulo en C de grado mínimo

- Nota: esta propiedad vale sólo en $\{0, 1\}$. Si no estamos en $\{0, 1\}$ hay que agregar la condición de que sea mónico para la unicidad.
- Prueba: Supongamos que hubiera dos distintos: $g_1 \neq g_2$.

Propiedad

Si C es lineal, entonces existe **un único** polinomio no nulo en C de grado mínimo

- Nota: esta propiedad vale sólo en $\{0, 1\}$. Si no estamos en $\{0, 1\}$ hay que agregar la condición de que sea mónico para la unicidad.
- Prueba: Supongamos que hubiera dos distintos: $g_1 \neq g_2$.
- Como son distintos, y estamos en $\{0, 1\}$, $g_1 + g_2 \neq 0$.

Propiedad

Si C es lineal, entonces existe **un único** polinomio no nulo en C de grado mínimo

- Nota: esta propiedad vale sólo en $\{0, 1\}$. Si no estamos en $\{0, 1\}$ hay que agregar la condición de que sea mónico para la unicidad.
- Prueba: Supongamos que hubiera dos distintos: $g_1 \neq g_2$.
- Como son distintos, y estamos en $\{0, 1\}$, $g_1 + g_2 \neq 0$.
- Como C es lineal, $g_1 + g_2$ está en C.

Propiedad

Si C es lineal, entonces existe **un único** polinomio no nulo en C de grado mínimo

- Nota: esta propiedad vale sólo en $\{0, 1\}$. Si no estamos en $\{0, 1\}$ hay que agregar la condición de que sea mónico para la unicidad.
- Prueba: Supongamos que hubiera dos distintos: $g_1 \neq g_2$.
- Como son distintos, y estamos en $\{0, 1\}$, $g_1 + g_2 \neq 0$.
- Como C es lineal, $g_1 + g_2$ está en C.
- Pero ¿cual es el grado de $g_1 + g_2$?

Continuación prueba

- Sea t el grado común a g_1, g_2.

Continuación prueba

- Sea t el grado común a g_1, g_2.
- Ambos son de la forma x^t+cosas de grado mas chico.

Continuación prueba

- Sea t el grado común a g_1, g_2.
- Ambos son de la forma x^t+cosas de grado mas chico.
- Por lo tanto, al sumarlos, queda $x^t + x^t+$cosas de grado mas chico.

Continuación prueba

- Sea t el grado común a g_1, g_2.
- Ambos son de la forma x^t+cosas de grado mas chico.
- Por lo tanto, al sumarlos, queda $x^t + x^t+$cosas de grado mas chico.
- Como estamos en $\{0, 1\}$, $x^t + x^t = 0$

Continuación prueba

- Sea t el grado común a g_1, g_2.
- Ambos son de la forma x^t+cosas de grado mas chico.
- Por lo tanto, al sumarlos, queda $x^t + x^t+$cosas de grado mas chico.
- Como estamos en $\{0, 1\}$, $x^t + x^t = 0$
- Asi que el grado de $g_1 + g_2$ es estrictamente menor que t

Continuación prueba

- Sea t el grado común a g_1, g_2.
- Ambos son de la forma x^t+cosas de grado mas chico.
- Por lo tanto, al sumarlos, queda $x^t + x^t+$cosas de grado mas chico.
- Como estamos en $\{0, 1\}$, $x^t + x^t = 0$
- Asi que el grado de $g_1 + g_2$ es estrictamente menor que t
- Absurdo, pues como $g_1 + g_2 \neq 0$, tendriamos un polinomio no nulo de grado mas chico que el menor grado de un polinomio no nulo.

Polinomio Generador

Definición

Si C es cíclico, el único polinomio no nulo de menor grado se llama el polinomio generador y se lo suele denotar por $g(x)$.

Polinomio Generador

Definición

Si C es cíclico, el único polinomio no nulo de menor grado se llama el polinomio generador y se lo suele denotar por $g(x)$.

- ¿Por qué se le llama el polinomio generador?

Polinomio Generador

Definición

Si C es cíclico, el único polinomio no nulo de menor grado se llama el polinomio generador y se lo suele denotar por $g(x)$.

- ¿Por qué se le llama el polinomio generador?
- Porque vamos a ver que el polinomio genera algebráicamente todo el código.

Polinomio Generador

Definición

Si C es cíclico, el único polinomio no nulo de menor grado se llama el polinomio generador y se lo suele denotar por $g(x)$.

- ¿Por qué se le llama el polinomio generador?
- Porque vamos a ver que el polinomio genera algebráicamente todo el código.
- Por lo tanto, en vez de tener que guardar una matriz generadora, basta con guardar al polinomio generador.

Polinomio Generador

Definición

Si C es cíclico, el único polinomio no nulo de menor grado se llama el polinomio generador y se lo suele denotar por $g(x)$.

- ¿Por qué se le llama el polinomio generador?
- Porque vamos a ver que el polinomio genera algebráicamente todo el código.
- Por lo tanto, en vez de tener que guardar una matriz generadora, basta con guardar al polinomio generador.
- De hecho, hay listas de códigos cíclicos muy utiles en la literatura, y lo único que se dan son los polinomios generadores.

Polinomio Generador

Definición

Si C es cíclico, el único polinomio no nulo de menor grado se llama el polinomio generador y se lo suele denotar por $g(x)$.

- ¿Por qué se le llama el polinomio generador?
- Porque vamos a ver que el polinomio genera algebráicamente todo el código.
- Por lo tanto, en vez de tener que guardar una matriz generadora, basta con guardar al polinomio generador.
- De hecho, hay listas de códigos cíclicos muy utiles en la literatura, y lo único que se dan son los polinomios generadores.
- De hecho, hay tablas de estos códigos, lo que se suele dar en cada entrada son los indices de los coeficientes que son 1.

Teorema fundamental de códigos cíclicos

Teorema

Sea $g(x)$ el polinomio generador de un código cíclico C de longitud n. Entonces:

Teorema fundamental de códigos cíclicos

Teorema

Sea $g(x)$ el polinomio generador de un código cíclico C de longitud n. Entonces:

1. C esta formado por los multiplos de $g(x)$ de grado menor que n: $C = \{p(x) : gr(p) < n \& g(x) | p(x)\}$

Teorema fundamental de códigos cíclicos

Teorema

Sea $g(x)$ el polinomio generador de un código cíclico C de longitud n. Entonces:

1. C esta formado por los multiplos de $g(x)$ de grado menor que n: $C = \{p(x) : gr(p) < n \& g(x) | p(x)\}$
2. $C = \{v(x) \odot g(x) : v \text{ es un polinomio cualquiera}\}$

Teorema

Sea $g(x)$ el polinomio generador de un código cíclico C de longitud n. Entonces:

1. C esta formado por los multiplos de $g(x)$ de grado menor que n:
   $C = \{p(x) : gr(p) < n \& g(x)|p(x)\}$
2. $C = \{v(x) \odot g(x) : v \text{ es un polinomio cualquiera}\}$
3. $gr(g(x)) = n - k$.

Teorema fundamental de códigos cíclicos

Teorema

Sea $g(x)$ el polinomio generador de un código cíclico C de longitud n. Entonces:

1. C esta formado por los multiplos de $g(x)$ de grado menor que n: $C = \{p(x) : gr(p) < n \& g(x) | p(x)\}$
2. $C = \{v(x) \odot g(x) : v \text{ es un polinomio cualquiera}\}$
3. $gr(g(x)) = n - k$.
4. $g(x)$ divide a $1 + x^n$

Teorema fundamental de códigos cíclicos

Teorema

Sea $g(x)$ el polinomio generador de un código cíclico C de longitud n. Entonces:

1. C esta formado por los multiplos de $g(x)$ de grado menor que n: $C = \{p(x) : gr(p) < n \& g(x)|p(x)\}$
2. $C = \{v(x) \odot g(x) : v \text{ es un polinomio cualquiera}\}$
3. $gr(g(x)) = n - k$.
4. $g(x)$ divide a $1 + x^n$
5. $g_0 = 1$

Prueba

- Prueba: Sea $C_1 = \{p(x) : gr(p) < n \& g(x) | p(x)\}$ y $C_2 = \{v(x) \odot g(x) : v \text{ es un polinomio cualquiera}\}$.

Prueba

- Prueba: Sea $C_1 = \{p(x) : gr(p) < n \& g(x) | p(x)\}$ y $C_2 = \{v(x) \odot g(x) : v \text{ es un polinomio cualquiera}\}$.
- Por la propiedad que probamos antes, $C_2 \subseteq C$, pues $g(x) \in C$.

Prueba

- Prueba: Sea $C_1 = \{p(x) : gr(p) < n \& g(x)|p(x)\}$ y $C_2 = \{v(x) \odot g(x) : v$ es un polinomio cualquiera$\}$.
- Por la propiedad que probamos antes, $C_2 \subseteq C$, pues $g(x) \in C$.
- Sea $p(x) \in C$.

Prueba

- Prueba: Sea $C_1 = \{p(x) : gr(p) < n \& g(x) | p(x)\}$ y $C_2 = \{v(x) \odot g(x) : v \text{ es un polinomio cualquiera}\}$.
- Por la propiedad que probamos antes, $C_2 \subseteq C$, pues $g(x) \in C$.
- Sea $p(x) \in C$.
- Dividamos $p(x)$ por $g(x)$, obteniendo polinomios $q(x)$ y $r(x)$, con $gr(r) < gr(g)$ tal que $p(x) = q(x)g(x) + r(x)$.

Prueba

- Prueba: Sea $C_1 = \{p(x) : gr(p) < n \& g(x)|p(x)\}$ y $C_2 = \{v(x) \odot g(x) : v \text{ es un polinomio cualquiera}\}$.
- Por la propiedad que probamos antes, $C_2 \subseteq C$, pues $g(x) \in C$.
- Sea $p(x) \in C$.
- Dividamos $p(x)$ por $g(x)$, obteniendo polinomios $q(x)$ y $r(x)$, con $gr(r) < gr(g)$ tal que $p(x) = q(x)g(x) + r(x)$.
- Por lo tanto $r(x) = p(x) + q(x)g(x)$.

Prueba

- Prueba: Sea $C_1 = \{p(x) : gr(p) < n \& g(x) | p(x)\}$ y $C_2 = \{v(x) \odot g(x) : v \text{ es un polinomio cualquiera}\}$.
- Por la propiedad que probamos antes, $C_2 \subseteq C$, pues $g(x) \in C$.
- Sea $p(x) \in C$.
- Dividamos $p(x)$ por $g(x)$, obteniendo polinomios $q(x)$ y $r(x)$, con $gr(r) < gr(g)$ tal que $p(x) = q(x)g(x) + r(x)$.
- Por lo tanto $r(x) = p(x) + q(x)g(x)$.
- Como $gr(r) < gr(g) < n$, entonces $r(x) = r(x) \bmod (1 + x^n)$

Prueba

- Prueba: Sea $C_1 = \{p(x) : gr(p) < n \& g(x) | p(x)\}$ y $C_2 = \{v(x) \odot g(x) : v \text{ es un polinomio cualquiera}\}$.
- Por la propiedad que probamos antes, $C_2 \subseteq C$, pues $g(x) \in C$.
- Sea $p(x) \in C$.
- Dividamos $p(x)$ por $g(x)$, obteniendo polinomios $q(x)$ y $r(x)$, con $gr(r) < gr(g)$ tal que $p(x) = q(x)g(x) + r(x)$.
- Por lo tanto $r(x) = p(x) + q(x)g(x)$.
- Como $gr(r) < gr(g) < n$, entonces $r(x) = r(x) \bmod (1 + x^n)$
- Como $p(x) \in C$, entonces $gr(p) < n$, y $p(x) = p(x) \bmod (1 + x^n)$

Prueba

- Prueba: Sea $C_1 = \{p(x) : gr(p) < n \& g(x) | p(x)\}$ y $C_2 = \{v(x) \odot g(x) : v \text{ es un polinomio cualquiera}\}$.
- Por la propiedad que probamos antes, $C_2 \subseteq C$, pues $g(x) \in C$.
- Sea $p(x) \in C$.
- Dividamos $p(x)$ por $g(x)$, obteniendo polinomios $q(x)$ y $r(x)$, con $gr(r) < gr(g)$ tal que $p(x) = q(x)g(x) + r(x)$.
- Por lo tanto $r(x) = p(x) + q(x)g(x)$.
- Como $gr(r) < gr(g) < n$, entonces $r(x) = r(x) \bmod (1 + x^n)$
- Como $p(x) \in C$, entonces $gr(p) < n$, y $p(x) = p(x) \bmod (1 + x^n)$
- Entonces:

Continuación Prueba

$$r(x) = r(x) \bmod (1 + x^n)$$

Continuación Prueba

$$\begin{aligned} r(x) &= r(x) \bmod (1 + x^n) \\ &= (p(x) + q(x)g(x)) \bmod (1 + x^n) \end{aligned}$$

Continuación Prueba

$$\begin{aligned}
r(x) &= r(x) \bmod (1 + x^n) \\
&= (p(x) + q(x)g(x)) \bmod (1 + x^n) \\
&= p(x) \bmod (1 + x^n) + (q(x)g(x)) \bmod (1 + x^n)
\end{aligned}$$

Continuación Prueba

$$
\begin{aligned}
r(x) &= r(x) \bmod (1 + x^n) \\
&= (p(x) + q(x)g(x)) \bmod (1 + x^n) \\
&= p(x) \bmod (1 + x^n) + (q(x)g(x)) \bmod (1 + x^n) \\
&= p(x) + q \odot g
\end{aligned}
$$

Continuación Prueba

$$
\begin{aligned}
r(x) &= r(x) \bmod (1+x^n) \\
&= (p(x)+q(x)g(x)) \bmod (1+x^n) \\
&= p(x) \bmod (1+x^n) + (q(x)g(x)) \bmod (1+x^n) \\
&= p(x) + q \odot g
\end{aligned}
$$

- Como $p(x) \in C$ y $q \odot g \in C$ y C es lineal, concluimos que $r \in C$.

Continuación Prueba

$$
\begin{aligned}
r(x) &= r(x) \bmod (1+x^n) \\
&= (p(x)+q(x)g(x)) \bmod (1+x^n) \\
&= p(x) \bmod (1+x^n) + (q(x)g(x)) \bmod (1+x^n) \\
&= p(x) + q \odot g
\end{aligned}
$$

- Como $p(x) \in C$ y $q \odot g \in C$ y C es lineal, concluimos que $r \in C$.
- Pero $gr(r) < gr(g)$ que es el polinomio no nulo de menor grado de C.

Continuación Prueba

$$
\begin{aligned}
r(x) &= r(x) \bmod (1+x^n) \\
&= (p(x)+q(x)g(x)) \bmod (1+x^n) \\
&= p(x) \bmod (1+x^n) + (q(x)g(x)) \bmod (1+x^n) \\
&= p(x) + q \odot g
\end{aligned}
$$

- Como $p(x) \in C$ y $q \odot g \in C$ y C es lineal, concluimos que $r \in C$.
- Pero $gr(r) < gr(g)$ que es el polinomio no nulo de menor grado de C.
- Concluimos que $r = 0$.

Continuación Prueba

$$\begin{aligned} r(x) &= r(x) \bmod (1+x^n) \\ &= (p(x)+q(x)g(x)) \bmod (1+x^n) \\ &= p(x) \bmod (1+x^n) + (q(x)g(x)) \bmod (1+x^n) \\ &= p(x) + q \odot g \end{aligned}$$

- Como $p(x) \in C$ y $q \odot g \in C$ y C es lineal, concluimos que $r \in C$.
- Pero $gr(r) < gr(g)$ que es el polinomio no nulo de menor grado de C.
- Concluimos que $r = 0$.
- Por lo tanto $p(x) = q(x)g(x) + r(x) = q(x)g(x) \in C_1$.

Continuación Prueba

- Entonces concluimos que $C \subseteq C_1$ y habiamos visto $C_2 \subseteq C$, sólo nos resta ver que $C_1 \subseteq C_2$.

Continuación Prueba

- Entonces concluimos que $C \subseteq C_1$ y habiamos visto $C_2 \subseteq C$, sólo nos resta ver que $C_1 \subseteq C_2$.
- Pero esa inclusión es obvia, pues si $gr(p) < n$ y $p(x) = q(x)g(x)$, entonces:

Continuación Prueba

- Entonces concluimos que $C \subseteq C_1$ y habiamos visto $C_2 \subseteq C$, sólo nos resta ver que $C_1 \subseteq C_2$.
- Pero esa inclusión es obvia, pues si $gr(p) < n$ y $p(x) = q(x)g(x)$, entonces:
- $p(x) = p(x) \bmod (1 + x^n)$ (pues $gr(p) < n$)

Continuación Prueba

- Entonces concluimos que $C \subseteq C_1$ y habiamos visto $C_2 \subseteq C$, sólo nos resta ver que $C_1 \subseteq C_2$.
- Pero esa inclusión es obvia, pues si $gr(p) < n$ y $p(x) = q(x)g(x)$, entonces:
- $p(x) = p(x) \bmod (1 + x^n)$ (pues $gr(p) < n$)
- Y por lo tanto $p(x) = q(x)g(x) \bmod (1 + x^n) = q \odot g \in C_2$.

Continuación Prueba

- Entonces concluimos que $C \subseteq C_1$ y habiamos visto $C_2 \subseteq C$, sólo nos resta ver que $C_1 \subseteq C_2$.
- Pero esa inclusión es obvia, pues si $gr(p) < n$ y $p(x) = q(x)g(x)$, entonces:
- $p(x) = p(x) \bmod (1 + x^n)$ (pues $gr(p) < n$)
- Y por lo tanto $p(x) = q(x)g(x) \bmod (1 + x^n) = q \odot g \in C_2$.
- Con esto hemos probado las partes 1) y 2) del teorema.

Continuación Prueba

- Entonces concluimos que $C \subseteq C_1$ y habiamos visto $C_2 \subseteq C$, sólo nos resta ver que $C_1 \subseteq C_2$.
- Pero esa inclusión es obvia, pues si $gr(p) < n$ y $p(x) = q(x)g(x)$, entonces:
- $p(x) = p(x) \bmod (1 + x^n)$ (pues $gr(p) < n$)
- Y por lo tanto $p(x) = q(x)g(x) \bmod (1 + x^n) = q \odot g \in C_2$.
- Con esto hemos probado las partes 1) y 2) del teorema.
- Vamos a la 3).

Continuación Prueba

- Sea t el grado de $g(x)$.

Continuación Prueba

- Sea t el grado de $g(x)$.
- Por 1), $p(x) \in C$ sii es de la forma $q(x)g(x)$ para algun polinomio $q(x)$.

Continuación Prueba

- Sea t el grado de $g(x)$.
- Por 1), $p(x) \in C$ sii es de la forma $q(x)g(x)$ para algun polinomio $q(x)$.
- Pero como el grado de los elementos de C es menor que n, entonces el grado de $q(x)g(x)$ debe ser menor que n.

Continuación Prueba

- Sea t el grado de $g(x)$.
- Por 1), $p(x) \in C$ sii es de la forma $q(x)g(x)$ para algun polinomio $q(x)$.
- Pero como el grado de los elementos de C es menor que n, entonces el grado de $q(x)g(x)$ debe ser menor que n.
- Por lo tanto el grado de $q(x)$ debe ser menor que $n - t$.

Continuación Prueba

- Sea t el grado de $g(x)$.
- Por 1), $p(x) \in C$ sii es de la forma $q(x)g(x)$ para algun polinomio $q(x)$.
- Pero como el grado de los elementos de C es menor que n, entonces el grado de $q(x)g(x)$ debe ser menor que n.
- Por lo tanto el grado de $q(x)$ debe ser menor que $n - t$.
- Asi, para cada polinomio de grado menor que $n - t$ corresponde un polinomio de C, y vicecersa.

Continuación Prueba

- Sea t el grado de $g(x)$.
- Por 1), $p(x) \in C$ sii es de la forma $q(x)g(x)$ para algun polinomio $q(x)$.
- Pero como el grado de los elementos de C es menor que n, entonces el grado de $q(x)g(x)$ debe ser menor que n.
- Por lo tanto el grado de $q(x)$ debe ser menor que $n - t$.
- Asi, para cada polinomio de grado menor que $n - t$ corresponde un polinomio de C, y vicecersa.
- Por lo tanto la cardinalidad de C es igual a la cardinalidad del conjunto de polinomios de grado menor que $n - t$.

- Sea t el grado de $g(x)$.
- Por 1), $p(x) \in C$ sii es de la forma $q(x)g(x)$ para algun polinomio $q(x)$.
- Pero como el grado de los elementos de C es menor que n, entonces el grado de $q(x)g(x)$ debe ser menor que n.
- Por lo tanto el grado de $q(x)$ debe ser menor que $n - t$.
- Asi, para cada polinomio de grado menor que $n - t$ corresponde un polinomio de C, y vicecersa.
- Por lo tanto la cardinalidad de C es igual a la cardinalidad del conjunto de polinomios de grado menor que $n - t$.
- ¿Cual es esa cardinalidad? Piensenlo un poco antes de ver la siguiente página.

Continuación Prueba

- Los polinomios de grado menor que $n-t$ tienen $n-t$ coeficientes (los de los términos de grado $0, 1, ..., n-t-1$).

Continuación Prueba

- Los polinomios de grado menor que $n - t$ tienen $n - t$ coeficientes (los de los términos de grado $0, 1, ..., n - t - 1$).
- Cada uno de esos coeficientes puede ser 1 o 0, asi que cada uno tiene dos posibilidades.

Continuación Prueba

- Los polinomios de grado menor que $n - t$ tienen $n - t$ coeficientes (los de los términos de grado $0, 1, ..., n - t - 1$).
- Cada uno de esos coeficientes puede ser 1 o 0, asi que cada uno tiene dos posibilidades.
- Como son $n - t$, el total de polinomios posibles es 2^{n-t}.

Continuación Prueba

- Los polinomios de grado menor que $n - t$ tienen $n - t$ coeficientes (los de los términos de grado $0, 1, ..., n - t - 1$).
- Cada uno de esos coeficientes puede ser 1 o 0, asi que cada uno tiene dos posibilidades.
- Como son $n - t$, el total de polinomios posibles es 2^{n-t}.
- Entonces hemos probado que la cardinalidad de C es 2^{n-t}.

Continuación Prueba

- Los polinomios de grado menor que $n - t$ tienen $n - t$ coeficientes (los de los términos de grado $0, 1, ..., n - t - 1$).
- Cada uno de esos coeficientes puede ser 1 o 0, asi que cada uno tiene dos posibilidades.
- Como son $n - t$, el total de polinomios posibles es 2^{n-t}.
- Entonces hemos probado que la cardinalidad de C es 2^{n-t}.
- Pero como C es lineal, sabemos que su cardinalidad es 2^k.

Continuación Prueba

- Los polinomios de grado menor que $n - t$ tienen $n - t$ coeficientes (los de los términos de grado $0, 1, ..., n - t - 1$).
- Cada uno de esos coeficientes puede ser 1 o 0, asi que cada uno tiene dos posibilidades.
- Como son $n - t$, el total de polinomios posibles es 2^{n-t}.
- Entonces hemos probado que la cardinalidad de C es 2^{n-t}.
- Pero como C es lineal, sabemos que su cardinalidad es 2^k.
- Asi que $2^k = 2^{n-t}$, por lo tanto $k = n - t$, y el grado de $g(x)$ es $t = n - k$.

Continuación Prueba

- Los polinomios de grado menor que $n-t$ tienen $n-t$ coeficientes (los de los términos de grado $0, 1, ..., n-t-1$).
- Cada uno de esos coeficientes puede ser 1 o 0, asi que cada uno tiene dos posibilidades.
- Como son $n-t$, el total de polinomios posibles es 2^{n-t}.
- Entonces hemos probado que la cardinalidad de C es 2^{n-t}.
- Pero como C es lineal, sabemos que su cardinalidad es 2^k.
- Asi que $2^k = 2^{n-t}$, por lo tanto $k = n-t$, y el grado de $g(x)$ es $t = n-k$.
- Fin parte 3

Continuación Prueba

- La parte 4) se puede probar de varias formas. Veamos una:

Continuación Prueba

- La parte 4) se puede probar de varias formas. Veamos una:
- Dividimos $1 + x^n$ por $g(x)$, obteniendo $q(x), r(x)$ con $gr(r) < gr(g)$ tal que $1 + x^n = q(x)g(x) + r(x)$.

Continuación Prueba

- La parte 4) se puede probar de varias formas. Veamos una:
- Dividimos $1 + x^n$ por $g(x)$, obteniendo $q(x), r(x)$ con $gr(r) < gr(g)$ tal que $1 + x^n = q(x)g(x) + r(x)$.
- Por lo tanto $r(x) = 1 + x^n + q(x)g(x)$.

- La parte 4) se puede probar de varias formas. Veamos una:
- Dividimos $1 + x^n$ por $g(x)$, obteniendo $q(x), r(x)$ con $gr(r) < gr(g)$ tal que $1 + x^n = q(x)g(x) + r(x)$.
- Por lo tanto $r(x) = 1 + x^n + q(x)g(x)$.
- Como $gr(r) < gr(g) < n$, $r(x) = r(x) \bmod (1 + x^n)$.

Continuación Prueba

- La parte 4) se puede probar de varias formas. Veamos una:
- Dividimos $1 + x^n$ por $g(x)$, obteniendo $q(x), r(x)$ con $gr(r) < gr(g)$ tal que $1 + x^n = q(x)g(x) + r(x)$.
- Por lo tanto $r(x) = 1 + x^n + q(x)g(x)$.
- Como $gr(r) < gr(g) < n$, $r(x) = r(x) \bmod (1 + x^n)$.
- Asi: $r(x) = (1 + x^n + q(x)g(x)) \bmod (1 + x^n) = q \odot g \in C$

Continuación Prueba

- La parte 4) se puede probar de varias formas. Veamos una:
- Dividimos $1 + x^n$ por $g(x)$, obteniendo $q(x), r(x)$ con $gr(r) < gr(g)$ tal que $1 + x^n = q(x)g(x) + r(x)$.
- Por lo tanto $r(x) = 1 + x^n + q(x)g(x)$.
- Como $gr(r) < gr(g) < n$, $r(x) = r(x) \bmod (1 + x^n)$.
- Asi: $r(x) = (1 + x^n + q(x)g(x)) \bmod (1 + x^n) = q \odot g \in C$
- (en la igualdad anterior usamos $(1 + x^n) \bmod (1 + x^n) = 0$)

Continuación Prueba

- La parte 4) se puede probar de varias formas. Veamos una:
- Dividimos $1 + x^n$ por $g(x)$, obteniendo $q(x), r(x)$ con $gr(r) < gr(g)$ tal que $1 + x^n = q(x)g(x) + r(x)$.
- Por lo tanto $r(x) = 1 + x^n + q(x)g(x)$.
- Como $gr(r) < gr(g) < n$, $r(x) = r(x) \bmod (1 + x^n)$.
- Asi: $r(x) = (1 + x^n + q(x)g(x)) \text{ mod } (1 + x^n) = q \odot g \in C$
- (en la igualdad anterior usamos $(1 + x^n) \bmod (1 + x^n) = 0$)
- Como $r \in C$ y $gr(r) < gr(g)$, entonces $r = 0$ y $g(x)|(1 + x^n)$

Continuación Prueba

- Otra prueba es observar que por la parte 3), g es de la forma $g_0 + g_1 x + ... + g_{n-k-1} x^{n-k-1} + x^{n-k}$

Continuación Prueba

- Otra prueba es observar que por la parte 3), g es de la forma $g_0 + g_1 x + ... + g_{n-k-1} x^{n-k-1} + x^{n-k}$
- Por lo tanto $x^k g(x) = g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + x^n$.

Continuación Prueba

- Otra prueba es observar que por la parte 3), g es de la forma $g_0 + g_1 x + ... + g_{n-k-1} x^{n-k-1} + x^{n-k}$
- Por lo tanto $x^k g(x) = g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + x^n$.
- Como 1+1=0, tenemos:

Continuación Prueba

- Otra prueba es observar que por la parte 3), g es de la forma $g_0 + g_1 x + ... + g_{n-k-1} x^{n-k-1} + x^{n-k}$
- Por lo tanto $x^k g(x) = g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + x^n$.
- Como 1+1=0, tenemos:
- $x^k g(x) = 1 + g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + (1 + x^n)$

Continuación Prueba

- Otra prueba es observar que por la parte 3), g es de la forma $g_0 + g_1 x + ... + g_{n-k-1} x^{n-k-1} + x^{n-k}$
- Por lo tanto $x^k g(x) = g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + x^n$.
- Como 1+1=0, tenemos:
- $x^k g(x) = 1 + g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + (1 + x^n)$
- Pero $1 + g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} = rot(g)$, asi que:

Continuación Prueba

- Otra prueba es observar que por la parte 3), g es de la forma $g_0 + g_1 x + ... + g_{n-k-1} x^{n-k-1} + x^{n-k}$
- Por lo tanto $x^k g(x) = g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + x^n$.
- Como 1+1=0, tenemos:
- $x^k g(x) = 1 + g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + (1 + x^n)$
- Pero $1 + g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} = rot(g)$, asi que:
- $x^k g(x) = rot(g) + (1 + x^n)$.

Continuación Prueba

- Otra prueba es observar que por la parte 3), g es de la forma $g_0 + g_1 x + ... + g_{n-k-1} x^{n-k-1} + x^{n-k}$
- Por lo tanto $x^k g(x) = g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + x^n$.
- Como 1+1=0, tenemos:
- $x^k g(x) = 1 + g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + (1 + x^n)$
- Pero $1 + g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} = rot(g)$, asi que:
- $x^k g(x) = rot(g) + (1 + x^n)$.
- Como $rot(g) \in C$, por la parte 1) del teorema tenemos que $rot(g) = q(x) g(x)$ para algún q de grado adecuado.

Continuación Prueba

- Otra prueba es observar que por la parte 3), g es de la forma $g_0 + g_1 x + ... + g_{n-k-1} x^{n-k-1} + x^{n-k}$
- Por lo tanto $x^k g(x) = g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + x^n$.
- Como 1+1=0, tenemos:
- $x^k g(x) = 1 + g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + (1 + x^n)$
- Pero $1 + g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} = rot(g)$, asi que:
- $x^k g(x) = rot(g) + (1 + x^n)$.
- Como $rot(g) \in C$, por la parte 1) del teorema tenemos que $rot(g) = q(x)g(x)$ para algún q de grado adecuado.
- Entonces $1 + x^n = x^k g(x) + rot(g) = x^k g(x) + q(x)g(x) = (x^k + q(x))g(x)$

Continuación Prueba

- Otra prueba es observar que por la parte 3), g es de la forma $g_0 + g_1 x + ... + g_{n-k-1} x^{n-k-1} + x^{n-k}$
- Por lo tanto $x^k g(x) = g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + x^n$.
- Como 1+1=0, tenemos:
- $x^k g(x) = 1 + g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} + (1 + x^n)$
- Pero $1 + g_0 x^k + g_1 x^{k+1} + ... + g_{n-k-1} x^{n-1} = rot(g)$, asi que:
- $x^k g(x) = rot(g) + (1 + x^n)$.
- Como $rot(g) \in C$, por la parte 1) del teorema tenemos que $rot(g) = q(x)g(x)$ para algún q de grado adecuado.
- Entonces $1 + x^n = x^k g(x) + rot(g) = x^k g(x) + q(x)g(x) = (x^k + q(x))g(x)$
- Es decir, g divide a $1 + x^n$.

Fin Prueba

- Para la parte 5) basta observar que si $1 + x^n = q(x)g(x)$ entonces $1 = q_0 g_0$ por lo tanto $g_0 = 1$.

Fin Prueba

- Para la parte 5) basta observar que si $1 + x^n = q(x)g(x)$ entonces $1 = q_0 g_0$ por lo tanto $g_0 = 1$.
- Fin prueba

Fin Prueba

- Para la parte 5) basta observar que si $1 + x^n = q(x)g(x)$ entonces $1 = q_0 g_0$ por lo tanto $g_0 = 1$.
- Fin prueba
- Como g divide a $1 + x^n$, $\frac{1+x^n}{g(x)}$ es un polinomio, que se suele llamar el polinomio chequeador y lo denotaremos por $h(x)$.

Fin Prueba

- Para la parte 5) basta observar que si $1 + x^n = q(x)g(x)$ entonces $1 = q_0 g_0$ por lo tanto $g_0 = 1$.
- Fin prueba
- Como g divide a $1 + x^n$, $\frac{1+x^n}{g(x)}$ es un polinomio, que se suele llamar el polinomio chequeador y lo denotaremos por $h(x)$.
- Se llama asi pues si $p(x) \in C$, entonces, como $p(x) = q(x)g(x)$ para algún q:

Fin Prueba

- Para la parte 5) basta observar que si $1 + x^n = q(x)g(x)$ entonces $1 = q_0 g_0$ por lo tanto $g_0 = 1$.
- Fin prueba
- Como g divide a $1 + x^n$, $\frac{1+x^n}{g(x)}$ es un polinomio, que se suele llamar el polinomio chequeador y lo denotaremos por $h(x)$.
- Se llama asi pues si $p(x) \in C$, entonces, como $p(x) = q(x)g(x)$ para algún q:
- $h(x) \odot p(x) = h(x)p(x) \bmod (1 + x^n) = h(x)q(x)g(x) \bmod (1 + x^n) = 0$.

Fin Prueba

- Para la parte 5) basta observar que si $1 + x^n = q(x)g(x)$ entonces $1 = q_0 g_0$ por lo tanto $g_0 = 1$.
- Fin prueba
- Como g divide a $1 + x^n$, $\frac{1+x^n}{g(x)}$ es un polinomio, que se suele llamar el polinomio chequeador y lo denotaremos por $h(x)$.
- Se llama asi pues si $p(x) \in C$, entonces, como $p(x) = q(x)g(x)$ para algún q:
- $h(x) \odot p(x) = h(x)p(x) \bmod (1 + x^n) = h(x)q(x)g(x) \bmod (1 + x^n) = 0$.
- La última igualdad pues $h(x)g(x) = 1 + x^n$ por definición de h.

Polinomio chequeador

- Viceversa, si p de grado $< n$ es tal que $h(x) \odot p(x) = 0$, entonces

Polinomio chequeador

- Viceversa, si p de grado $< n$ es tal que $h(x) \odot p(x) = 0$, entonces
- $h(x)p(x) \bmod (1 + x^n) = 0$, es decir $1 + x^n$ divide a $h(x)p(x)$.

Polinomio chequeador

- Viceversa, si p de grado $< n$ es tal que $h(x) \odot p(x) = 0$, entonces
- $h(x)p(x) \bmod (1 + x^n) = 0$, es decir $1 + x^n$ divide a $h(x)p(x)$.
- Por lo tanto existe $q(x)$ con $h(x)p(x) = (1 + x^n)q(x)$.

Polinomio chequeador

- Viceversa, si p de grado $< n$ es tal que $h(x) \odot p(x) = 0$, entonces
- $h(x)p(x) \bmod (1 + x^n) = 0$, es decir $1 + x^n$ divide a $h(x)p(x)$.
- Por lo tanto existe $q(x)$ con $h(x)p(x) = (1 + x^n)q(x)$.
- Pero $1 + x^n = h(x)g(x)$ asi que $h(x)p(x) = h(x)g(x)q(x)$

Polinomio chequeador

- Viceversa, si p de grado $< n$ es tal que $h(x) \odot p(x) = 0$, entonces
- $h(x)p(x) \bmod (1 + x^n) = 0$, es decir $1 + x^n$ divide a $h(x)p(x)$.
- Por lo tanto existe $q(x)$ con $h(x)p(x) = (1 + x^n)q(x)$.
- Pero $1 + x^n = h(x)g(x)$ asi que $h(x)p(x) = h(x)g(x)q(x)$
- Simplificando h tenemos que $p(x) = q(x)g(x)$ y por lo tanto $p \in C$.

Polinomio chequeador

- Viceversa, si p de grado $< n$ es tal que $h(x) \odot p(x) = 0$, entonces
- $h(x)p(x) \bmod (1 + x^n) = 0$, es decir $1 + x^n$ divide a $h(x)p(x)$.
- Por lo tanto existe $q(x)$ con $h(x)p(x) = (1 + x^n)q(x)$.
- Pero $1 + x^n = h(x)g(x)$ asi que $h(x)p(x) = h(x)g(x)q(x)$
- Simplificando h tenemos que $p(x) = q(x)g(x)$ y por lo tanto $p \in C$.
- Asi que podemos “chequear" si un polinomio está en C o no “multiplicando" (modulo $1 + x^n$) por $h(x)$ y viendo si da 0 o no

Sobre los ejercicios

- En los ejercicios, el polinomio generador se los daremos nosotros

Sobre los ejercicios

- En los ejercicios, el polinomio generador se los daremos nosotros
- Es decir, no es que les demos un código y les vamos a pedir que calculen el polinomio generador (bueno, podria ser, pero sólo si es un código con pocas palabras) sino que les vamos a dar $g(x)$ y el n, y les vamos a pedir que hagan varias cosas a partir de ellos.

Sobre los ejercicios

- En los ejercicios, el polinomio generador se los daremos nosotros
- Es decir, no es que les demos un código y les vamos a pedir que calculen el polinomio generador (bueno, podria ser, pero sólo si es un código con pocas palabras) sino que les vamos a dar $g(x)$ y el n, y les vamos a pedir que hagan varias cosas a partir de ellos.
- Pej, calcular h, o la dimensión de C.

Sobre los ejercicios

- En los ejercicios, el polinomio generador se los daremos nosotros
- Es decir, no es que les demos un código y les vamos a pedir que calculen el polinomio generador (bueno, podria ser, pero sólo si es un código con pocas palabras) sino que les vamos a dar $g(x)$ y el n, y les vamos a pedir que hagan varias cosas a partir de ellos.
- Pej, calcular h, o la dimensión de C.
- O dar matrices generadoras para el código, o codificar/decodificar palabras, usando algunos de los métodos que vienen a continuación.

Primer método de codificación

- Este teorema da lugar a dos formas de codificar y decodificar palabras.

Primer método de codificación

- Este teorema da lugar a dos formas de codificar y decodificar palabras.
- Recordemos que por "codificar" entendemos el proceso de tomar las palabras de $\{0, 1\}^k$ y a cada una de ellas asignarle una palabra de C

Primer método de codificación

- Este teorema da lugar a dos formas de codificar y decodificar palabras.
- Recordemos que por “codificar" entendemos el proceso de tomar las palabras de $\{0,1\}^k$ y a cada una de ellas asignarle una palabra de C
- El primer método usa directamente la propiedad 1).

Primer método de codificación

- Este teorema da lugar a dos formas de codificar y decodificar palabras.
- Recordemos que por “codificar" entendemos el proceso de tomar las palabras de $\{0,1\}^k$ y a cada una de ellas asignarle una palabra de C
- El primer método usa directamente la propiedad 1).
- Es decir, dada una palabra en $\{0,1\}^k$, la cual estará identificada con un polinomio u de grado menor a k, la palabra asociada en C es simplemente $u(x)g(x)$.

Primer método de codificación

- Este teorema da lugar a dos formas de codificar y decodificar palabras.
- Recordemos que por “codificar" entendemos el proceso de tomar las palabras de $\{0,1\}^k$ y a cada una de ellas asignarle una palabra de C
- El primer método usa directamente la propiedad 1).
- Es decir, dada una palabra en $\{0,1\}^k$, la cual estará identificada con un polinomio u de grado menor a k, la palabra asociada en C es simplemente $u(x)g(x)$.
- (producto usual, pues $gr(u(x)g(x)) = gr(u) + gr(g) < k + n - k = n$).

- Ejemplo: Sea C el código con longitud $n = 7$ y polinomio generador $g(x) = 1 + x^2 + x^3$, que corresponde a la palabra 1011000

Primer método de codificación

- Ejemplo: Sea C el código con longitud $n = 7$ y polinomio generador $g(x) = 1 + x^2 + x^3$, que corresponde a la palabra 1011000
- La dimensión de C, de acuerdo con el teorema, es $n = 7 - 3 = 4$.

Primer método de codificación

- Ejemplo: Sea C el código con longitud $n = 7$ y polinomio generador $g(x) = 1 + x^2 + x^3$, que corresponde a la palabra 1011000
- La dimensión de C, de acuerdo con el teorema, es $n = 7 - 3 = 4$.
- Por lo tanto C tiene $2^4 = 16$ palabras.

Primer método de codificación

- Ejemplo: Sea C el código con longitud $n = 7$ y polinomio generador $g(x) = 1 + x^2 + x^3$, que corresponde a la palabra 1011000
- La dimensión de C, de acuerdo con el teorema, es $n = 7 - 3 = 4$.
- Por lo tanto C tiene $2^4 = 16$ palabras.
- Supongamos que queremos codificar la palabra $0110 \in \{0, 1\}^4$.

Primer método de codificación

- Ejemplo: Sea C el código con longitud $n = 7$ y polinomio generador $g(x) = 1 + x^2 + x^3$, que corresponde a la palabra 1011000
- La dimensión de C, de acuerdo con el teorema, es $n = 7 - 3 = 4$.
- Por lo tanto C tiene $2^4 = 16$ palabras.
- Supongamos que queremos codificar la palabra $0110 \in \{0, 1\}^4$.
- Corresponde al polinomio $x + x^2$.

Primer método de codificación

- Ejemplo: Sea C el código con longitud $n = 7$ y polinomio generador $g(x) = 1 + x^2 + x^3$, que corresponde a la palabra 1011000
- La dimensión de C, de acuerdo con el teorema, es $n = 7 - 3 = 4$.
- Por lo tanto C tiene $2^4 = 16$ palabras.
- Supongamos que queremos codificar la palabra $0110 \in \{0,1\}^4$.
- Corresponde al polinomio $x + x^2$.
- Usando el primer método, simplemente hacemos

$$(x+x^2)(1+x^2+x^3) = x+x^3+x^4+x^2+x^4+x^5 = x+x^2+x^3+x^5$$

Primer método de codificación

- Ejemplo: Sea C el código con longitud $n = 7$ y polinomio generador $g(x) = 1 + x^2 + x^3$, que corresponde a la palabra 1011000
- La dimensión de C, de acuerdo con el teorema, es $n = 7 - 3 = 4$.
- Por lo tanto C tiene $2^4 = 16$ palabras.
- Supongamos que queremos codificar la palabra $0110 \in \{0, 1\}^4$.
- Corresponde al polinomio $x + x^2$.
- Usando el primer método, simplemente hacemos

$$(x+x^2)(1+x^2+x^3) = x+x^3+x^4+x^2+x^4+x^5 = x+x^2+x^3+x^5$$

- Que corresponde a la palabra 0111010.

Primer método de codificación

- Aparentemente (yo no sé de esto, ustedes deben saberlo de Organización/Arquitectura de computadoras) multiplicar polinomios es algo que se “programa" fácilmente en hardware y es muy rápido

Primer método de codificación

- Aparentemente (yo no sé de esto, ustedes deben saberlo de Organización/Arquitectura de computadoras) multiplicar polinomios es algo que se “programa" fácilmente en hardware y es muy rápido
- En software es mas díficil pero pej tengo entendido que en los chips de Intel vienen instrucciones especiales para realizar esto mas fácilmente.

Primer método de codificación

- Aparentemente (yo no sé de esto, ustedes deben saberlo de Organización/Arquitectura de computadoras) multiplicar polinomios es algo que se “programa" fácilmente en hardware y es muy rápido
- En software es mas díficil pero pej tengo entendido que en los chips de Intel vienen instrucciones especiales para realizar esto mas fácilmente.
- Un problema con este método es la decodificación.

Primer método de codificación

- Aparentemente (yo no sé de esto, ustedes deben saberlo de Organización/Arquitectura de computadoras) multiplicar polinomios es algo que se “programa" fácilmente en hardware y es muy rápido
- En software es mas díficil pero pej tengo entendido que en los chips de Intel vienen instrucciones especiales para realizar esto mas fácilmente.
- Un problema con este método es la decodificación.
- Observemos que la palabra codificada 0110 no “aparece" en la palabra código 0111010

Primer método de codificación

- Aparentemente (yo no sé de esto, ustedes deben saberlo de Organización/Arquitectura de computadoras) multiplicar polinomios es algo que se “programa" fácilmente en hardware y es muy rápido
- En software es mas díficil pero pej tengo entendido que en los chips de Intel vienen instrucciones especiales para realizar esto mas fácilmente.
- Un problema con este método es la decodificación.
- Observemos que la palabra codificada 0110 no “aparece" en la palabra código 0111010
- Esto ocurre en general, salvo casualidad.

Primer método de codificación

- Aparentemente (yo no sé de esto, ustedes deben saberlo de Organización/Arquitectura de computadoras) multiplicar polinomios es algo que se “programa" fácilmente en hardware y es muy rápido
- En software es mas díficil pero pej tengo entendido que en los chips de Intel vienen instrucciones especiales para realizar esto mas fácilmente.
- Un problema con este método es la decodificación.
- Observemos que la palabra codificada 0110 no “aparece" en la palabra código 0111010
- Esto ocurre en general, salvo casualidad.
- ¿Por qué?

Matriz generadora correspondiente al primer método de codificación

- Supongamos que codificamos 10...0, 01...0, etc de $\{0, 1\}^k$.

Matriz generadora correspondiente al primer método de codificación

- Supongamos que codificamos 10...0, 01...0, etc de $\{0,1\}^k$.
- Es decir, queremos codificar $1, x, ..., x^{k-1}$.

Matriz generadora correspondiente al primer método de codificación

- Supongamos que codificamos 10...0, 01...0, etc de $\{0,1\}^k$.
- Es decir, queremos codificar $1, x, ..., x^{k-1}$.
- Las palabras codificadas serán $g(x)$,$xg(x)$,...,$x^{k-1}g(x)$

Matriz generadora correspondiente al primer método de codificación

- Supongamos que codificamos 10...0, 01...0, etc de $\{0,1\}^k$.
- Es decir, queremos codificar $1, x, ..., x^{k-1}$.
- Las palabras codificadas serán $g(x)$,$xg(x)$,...,$x^{k-1}g(x)$
- Las cuales son claramente LI pues los grados son todos distintos.

Matriz generadora correspondiente al primer método de codificación

- Supongamos que codificamos 10...0, 01...0, etc de $\{0,1\}^k$.
- Es decir, queremos codificar $1, x, ..., x^{k-1}$.
- Las palabras codificadas serán $g(x)$,$xg(x)$,...,$x^{k-1}g(x)$
- Las cuales son claramente LI pues los grados son todos distintos.
- Es decir, $\{g(x), xg(x), ..., x^{k-1}g(x)\}$ es una BASE de C.

Matriz generadora correspondiente al primer método de codificación

- Supongamos que codificamos 10...0, 01...0, etc de $\{0, 1\}^k$.
- Es decir, queremos codificar $1, x, ..., x^{k-1}$.
- Las palabras codificadas serán $g(x)$,$xg(x)$,...,$x^{k-1}g(x)$
- Las cuales son claramente LI pues los grados son todos distintos.
- Es decir, $\{g(x), xg(x), ..., x^{k-1}g(x)\}$ es una BASE de C.
- Esto da una matriz generadora, que tiene la forma: (recordemos que $g_0 = 1 = g_{n-k}$

Matriz generadora correspondiente al primer método de codificación

$$G = \begin{bmatrix} 1 & g_1 & \dots & \dots & g_{n-k-1} & 1 & 0 & \dots & 0 \\ 0 & 1 & g_1 & \dots & \dots & g_{n-k-1} & 1 & \dots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 1\dots & 0 \\ 0 & 0 & \dots & 1 & g_1 & \dots & \dots & g_{n-k-1} & 1 \end{bmatrix}$$

Matriz generadora correspondiente al primer método de codificación

Por ejemplo, con $g(x) = 1 + x^2 + x^3$ y $n = 7$:

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

Esta matriz no “tiene" la identidad en ningún lado.

Matriz generadora correspondiente al primer método de codificación

- Por eso decodificar no es tan fácil como cuando se tiene una matriz generadora con la identidad.

Matriz generadora correspondiente al primer método de codificación

- Por eso decodificar no es tan fácil como cuando se tiene una matriz generadora con la identidad.
- Para decodificar una palabra, hay que dividirla por $g(x)$.

Matriz generadora correspondiente al primer método de codificación

- Por eso decodificar no es tan fácil como cuando se tiene una matriz generadora con la identidad.
- Para decodificar una palabra, hay que dividirla por $g(x)$.
- Esto tambien se hace fácil en hardware, pero no tan fácil en software.

Matriz generadora correspondiente al primer método de codificación

- Por eso decodificar no es tan fácil como cuando se tiene una matriz generadora con la identidad.
- Para decodificar una palabra, hay que dividirla por $g(x)$.
- Esto tambien se hace fácil en hardware, pero no tan fácil en software.
- Por eso el segundo método que daremos, menos intuitivo que el primero, es preferible, pues da origen a una matriz generadora que si tiene la identidad, haciendo que decodificar sea muy fácil.

Segundo método de codificación

- Por el teorema, los elementos de C son los múltiplos de g de grado menor que n.

Segundo método de codificación

- Por el teorema, los elementos de C son los múltiplos de g de grado menor que n.
- Dado un polinomio cualquiera $p(x)$ de grado menor que n, observemos que:

Segundo método de codificación

- Por el teorema, los elementos de C son los múltiplos de g de grado menor que n.
- Dado un polinomio cualquiera $p(x)$ de grado menor que n, observemos que:
  - $(p(x) \bmod g(x)) + p(x)$ es múltiplo de g !

Segundo método de codificación

- Por el teorema, los elementos de C son los múltiplos de g de grado menor que n.
- Dado un polinomio cualquiera $p(x)$ de grado menor que n, observemos que:
  - $(p(x) \bmod g(x)) + p(x)$ es múltiplo de g !
- Pues por definición, $(p(x) \bmod g(x))$ es el resto de dividir p por g, es decir, existe q tal que $p(x) = q(x)g(x) + (p(x) \bmod g(x))$

Segundo método de codificación

- Por el teorema, los elementos de C son los múltiplos de g de grado menor que n.
- Dado un polinomio cualquiera $p(x)$ de grado menor que n, observemos que:
  - $(p(x) \bmod g(x)) + p(x)$ es múltiplo de g !
- Pues por definición, $(p(x) \bmod g(x))$ es el resto de dividir p por g, es decir, existe q tal que $p(x) = q(x)g(x) + (p(x) \bmod g(x))$
- Por lo tanto $(p(x) \bmod g(x)) + p(x) = q(x)g(x)$ es un múltiplo de g.

Segundo método de codificación

- Por el teorema, los elementos de C son los múltiplos de g de grado menor que n.
- Dado un polinomio cualquiera $p(x)$ de grado menor que n, observemos que:
  - $(p(x) \bmod g(x)) + p(x)$ es múltiplo de g !
- Pues por definición, $(p(x) \bmod g(x))$ es el resto de dividir p por g, es decir, existe q tal que $p(x) = q(x)g(x) + (p(x) \bmod g(x))$
- Por lo tanto $(p(x) \bmod g(x)) + p(x) = q(x)g(x)$ es un múltiplo de g.
- Asi que en vez de codificar una palabra multiplicandola por g, podemos usar este truco de arriba.

Segundo método de codificación

- Por el teorema, los elementos de C son los múltiplos de g de grado menor que n.
- Dado un polinomio cualquiera $p(x)$ de grado menor que n, observemos que:
  - $(p(x) \bmod g(x)) + p(x)$ es múltiplo de g !
- Pues por definición, $(p(x) \bmod g(x))$ es el resto de dividir p por g, es decir, existe q tal que $p(x) = q(x)g(x) + (p(x) \bmod g(x))$
- Por lo tanto $(p(x) \bmod g(x)) + p(x) = q(x)g(x)$ es un múltiplo de g.
- Asi que en vez de codificar una palabra multiplicandola por g, podemos usar este truco de arriba.
- Pero hay que tener cuidado.

Segundo método de codificación

- Lo primero que uno pensaria es decir, “bueno, dada una palabra $u \in \{0,1\}^k$, la miro como polinomio $u(x)$ y la codifico como $(u(x) \mod g(x)) + u(x)$"

Segundo método de codificación

- Lo primero que uno pensaria es decir, “bueno, dada una palabra $u \in \{0,1\}^k$, la miro como polinomio $u(x)$ y la codifico como $(u(x) \bmod g(x)) + u(x)$"
- Pero esto esta MAL.

Segundo método de codificación

- Lo primero que uno pensaria es decir, “bueno, dada una palabra $u \in \{0, 1\}^k$, la miro como polinomio $u(x)$ y la codifico como ($u(x)$ mod $g(x)) + u(x)$"
- Pero esto esta MAL.
- Cuando uno codifica una palabra u asignandole una palabra v del código, el procedimiento para asignar $u \mapsto v$ debe ser tal que a dos u distintas se les asigne dos v distintos, si no luego no se puede decodificar.

Segundo método de codificación

- Lo primero que uno pensaria es decir, “bueno, dada una palabra $u \in \{0,1\}^k$, la miro como polinomio $u(x)$ y la codifico como $(u(x) \bmod g(x)) + u(x)$"
- Pero esto esta MAL.
- Cuando uno codifica una palabra u asignandole una palabra v del código, el procedimiento para asignar $u \mapsto v$ debe ser tal que a dos u distintas se les asigne dos v distintos, si no luego no se puede decodificar.
- Y la función $u(x) \mapsto (u(x) \bmod g(x)) + u(x)$ no es inyectiva, no cumple con esa propiedad.

Segundo método de codificación

- Lo primero que uno pensaria es decir, “bueno, dada una palabra $u \in \{0,1\}^k$, la miro como polinomio $u(x)$ y la codifico como $(u(x) \bmod g(x)) + u(x)$"
- Pero esto esta MAL.
- Cuando uno codifica una palabra u asignandole una palabra v del código, el procedimiento para asignar $u \mapsto v$ debe ser tal que a dos u distintas se les asigne dos v distintos, si no luego no se puede decodificar.
- Y la función $u(x) \mapsto (u(x) \bmod g(x)) + u(x)$ no es inyectiva, no cumple con esa propiedad.
- Ejemplo fácil: Si $k \leq n-k$, entonces:

Segundo método de codificación

- Lo primero que uno pensaria es decir, “bueno, dada una palabra $u \in \{0, 1\}^k$, la miro como polinomio $u(x)$ y la codifico como $(u(x) \mod g(x)) + u(x)$"
- Pero esto esta MAL.
- Cuando uno codifica una palabra u asignandole una palabra v del código, el procedimiento para asignar $u \mapsto v$ debe ser tal que a dos u distintas se les asigne dos v distintos, si no luego no se puede decodificar.
- Y la función $u(x) \mapsto (u(x) \mod g(x)) + u(x)$ no es inyectiva, no cumple con esa propiedad.
- Ejemplo fácil: Si $k \leq n - k$, entonces:
- $(u(x) \mod g(x)) + u(x) = u(x) + u(x) = 0$ para todo $u(x)$ de grado menor que k!!!!

Segundo método de codificación

- ¿Y entonces?

Segundo método de codificación

- ¿Y entonces?
- Entonces, un trick: primero codificamos $u(x)$ con un $p(x)$ que asegure que $u(x)(\mapsto p(x)) \mapsto (p(x) \bmod g(x)) + p(x)$ sea inyectiva.

Segundo método de codificación

- ¿Y entonces?
- Entonces, un trick: primero codificamos $u(x)$ con un $p(x)$ que asegure que $u(x)(\mapsto p(x)) \mapsto (p(x) \bmod g(x)) + p(x)$ sea inyectiva.
- Tomaremos $p(x) = u(x)x^{n-k}$.

Segundo método de codificación

- ¿Y entonces?
- Entonces, un trick: primero codificamos $u(x)$ con un $p(x)$ que asegure que $u(x)(\mapsto p(x)) \mapsto (p(x) \bmod g(x)) + p(x)$ sea inyectiva.
- Tomaremos $p(x) = u(x)x^{n-k}$.
- Como $gr(u) < k$, entonces $gr(p) < n$.

Segundo método de codificación

- ¿Y entonces?
- Entonces, un trick: primero codificamos $u(x)$ con un $p(x)$ que asegure que $u(x)(\mapsto p(x)) \mapsto (p(x) \bmod g(x)) + p(x)$ sea inyectiva.
- Tomaremos $p(x) = u(x)x^{n-k}$.
- Como $gr(u) < k$, entonces $gr(p) < n$.
- Supongamos que $u \neq w$ pero que:

$$(u(x)x^{n-k} \bmod g(x)) + u(x)x^{n-k} = (w(x)x^{n-k} \bmod g(x)) + w(x)x^{n-k}$$

Segundo método de codificación

- ¿Y entonces?
- Entonces, un trick: primero codificamos $u(x)$ con un $p(x)$ que asegure que $u(x)(\mapsto p(x)) \mapsto (p(x) \bmod g(x)) + p(x)$ sea inyectiva.
- Tomaremos $p(x) = u(x)x^{n-k}$.
- Como $gr(u) < k$, entonces $gr(p) < n$.
- Supongamos que $u \neq w$ pero que:

$$(u(x)x^{n-k} \bmod g(x)) + u(x)x^{n-k} = (w(x)x^{n-k} \bmod g(x)) + w(x)x^{n-k}$$

- Luego: $(u(x) + w(x)))\, x^{n-k} = (u(x) + w(x))x^{n-k} \bmod g(x))$.

Segundo método de codificación

- ¿Y entonces?
- Entonces, un trick: primero codificamos $u(x)$ con un $p(x)$ que asegure que $u(x)(\mapsto p(x)) \mapsto (p(x) \bmod g(x)) + p(x)$ sea inyectiva.
- Tomaremos $p(x) = u(x)x^{n-k}$.
- Como $gr(u) < k$, entonces $gr(p) < n$.
- Supongamos que $u \neq w$ pero que:

$$(u(x)x^{n-k} \bmod g(x)) + u(x)x^{n-k} = (w(x)x^{n-k} \bmod g(x)) + w(x)x^{n-k}$$

- Luego: $(u(x) + w(x)))\, x^{n-k} = (u(x) + w(x))x^{n-k} \bmod g(x))$.
- Pero el polinomio de la derecha tiene grado menor que $gr(g) = n - k$, mientras que el polinomio de la izquierda tiene grado mayor o igual a $n - k$, absurdo.

Segundo método de codificación

- Entonces este método sirve para codificar.

Segundo método de codificación

- Entonces este método sirve para codificar.
- Mas aún, justamente como en $(u(x)x^{n-k} \bmod g(x)) + u(x)x^{n-k}$ la parte $u(x)x^{n-k}$ tiene grado mayor o igual que $n-k$ mientras que $(u(x)x^{n-k} \bmod g(x))$ tiene grado menor que $gr(g) = n-k$ (esto es lo que usamos en la pag. anterior para probar inyectividad) entonces la parte $u(x)x^{n-k}$ queda inalterada por la parte $(u(x)x^{n-k} \bmod g(x))$

Segundo método de codificación

- Entonces este método sirve para codificar.
- Mas aún, justamente como en $(u(x)x^{n-k} \bmod g(x)) + u(x)x^{n-k}$ la parte $u(x)x^{n-k}$ tiene grado mayor o igual que $n-k$ mientras que $(u(x)x^{n-k} \bmod g(x))$ tiene grado menor que $gr(g) = n-k$ (esto es lo que usamos en la pag. anterior para probar inyectividad) entonces la parte $u(x)x^{n-k}$ queda inalterada por la parte $(u(x)x^{n-k} \bmod g(x))$
- Por lo tanto mirando los coeficientes de grado mayor o igual a $n-k$, podemos recuperar $u(x)x^{n-k}$ y de ahi recuperar $u(x)$.

Segundo método de codificación

- Entonces este método sirve para codificar.
- Mas aún, justamente como en $(u(x)x^{n-k} \mod g(x)) + u(x)x^{n-k}$ la parte $u(x)x^{n-k}$ tiene grado mayor o igual que $n-k$ mientras que $(u(x)x^{n-k} \mod g(x))$ tiene grado menor que $gr(g) = n-k$ (esto es lo que usamos en la pag. anterior para probar inyectividad) entonces la parte $u(x)x^{n-k}$ queda inalterada por la parte $(u(x)x^{n-k} \mod g(x))$
- Por lo tanto mirando los coeficientes de grado mayor o igual a $n-k$, podemos recuperar $u(x)x^{n-k}$ y de ahi recuperar $u(x)$.
- Asi que decodificar es muy fácil.

Segundo método de codificación

- Entonces este método sirve para codificar.
- Mas aún, justamente como en $(u(x)x^{n-k} \bmod g(x)) + u(x)x^{n-k}$ la parte $u(x)x^{n-k}$ tiene grado mayor o igual que $n-k$ mientras que $(u(x)x^{n-k} \bmod g(x))$ tiene grado menor que $gr(g) = n-k$ (esto es lo que usamos en la pag. anterior para probar inyectividad) entonces la parte $u(x)x^{n-k}$ queda inalterada por la parte $(u(x)x^{n-k} \bmod g(x))$
- Por lo tanto mirando los coeficientes de grado mayor o igual a $n-k$, podemos recuperar $u(x)x^{n-k}$ y de ahi recuperar $u(x)$.
- Asi que decodificar es muy fácil.
- Veamos un ejemplo.

Segundo método de codificación: Ejemplo

- Tomemos como antes $n = 7$, polinomio generador $g(x) = 1 + x^2 + x^3$ y $u(x) = 0110 = x + x^2$

Segundo método de codificación: Ejemplo

- Tomemos como antes $n = 7$, polinomio generador $g(x) = 1 + x^2 + x^3$ y $u(x) = 0110 = x + x^2$
- $u(x)x^{n-k} = (x + x^2)x^3 = x^4 + x^5$

Segundo método de codificación: Ejemplo

- Tomemos como antes $n = 7$, polinomio generador $g(x) = 1 + x^2 + x^3$ y $u(x) = 0110 = x + x^2$
- $u(x)x^{n-k} = (x + x^2)x^3 = x^4 + x^5$
- Debemos calcular $(x^4 + x^5) \bmod g(x)$.

Segundo método de codificación: Ejemplo

- Tomemos como antes $n = 7$, polinomio generador $g(x) = 1 + x^2 + x^3$ y $u(x) = 0110 = x + x^2$
- $u(x)x^{n-k} = (x + x^2)x^3 = x^4 + x^5$
- Debemos calcular $(x^4 + x^5) \bmod g(x)$.
- En principio debemos dividir $x^4 + x^5$ por $g(x)$ y obtener el resto, pero hay una forma mas fácil.

Segundo método de codificación: Ejemplo

- Tomemos como antes $n = 7$, polinomio generador $g(x) = 1 + x^2 + x^3$ y $u(x) = 0110 = x + x^2$
- $u(x)x^{n-k} = (x + x^2)x^3 = x^4 + x^5$
- Debemos calcular $(x^4 + x^5) \bmod g(x)$.
- En principio debemos dividir $x^4 + x^5$ por $g(x)$ y obtener el resto, pero hay una forma mas fácil.
- Ciertamente $g(x) \bmod g(x) = 0$.

Segundo método de codificación: Ejemplo

- Tomemos como antes $n = 7$, polinomio generador $g(x) = 1 + x^2 + x^3$ y $u(x) = 0110 = x + x^2$
- $u(x)x^{n-k} = (x + x^2)x^3 = x^4 + x^5$
- Debemos calcular $(x^4 + x^5) \bmod g(x)$.
- En principio debemos dividir $x^4 + x^5$ por $g(x)$ y obtener el resto, pero hay una forma mas fácil.
- Ciertamente $g(x) \bmod g(x) = 0$.
- Es decir $(1 + x^2 + x^3) \bmod g(x) = 0$.

Segundo método de codificación: Ejemplo

- Tomemos como antes $n = 7$, polinomio generador $g(x) = 1 + x^2 + x^3$ y $u(x) = 0110 = x + x^2$
- $u(x)x^{n-k} = (x + x^2)x^3 = x^4 + x^5$
- Debemos calcular $(x^4 + x^5) \bmod g(x)$.
- En principio debemos dividir $x^4 + x^5$ por $g(x)$ y obtener el resto, pero hay una forma mas fácil.
- Ciertamente $g(x) \bmod g(x) = 0$.
- Es decir $(1 + x^2 + x^3) \bmod g(x) = 0$.
- Por otro lado, como $gr(1 + x^2) < gr(g)$ entonces tenemos que $(1 + x^2 + x^3) \bmod g(x) = 1 + x^2 + \left(x^3 \bmod g(x)\right)$.

Segundo método de codificación: Ejemplo

- Tomemos como antes $n = 7$, polinomio generador $g(x) = 1 + x^2 + x^3$ y $u(x) = 0110 = x + x^2$
- $u(x)x^{n-k} = (x + x^2)x^3 = x^4 + x^5$
- Debemos calcular $(x^4 + x^5) \bmod g(x)$.
- En principio debemos dividir $x^4 + x^5$ por $g(x)$ y obtener el resto, pero hay una forma mas fácil.
- Ciertamente $g(x) \bmod g(x) = 0$.
- Es decir $(1 + x^2 + x^3) \bmod g(x) = 0$.
- Por otro lado, como $gr(1 + x^2) < gr(g)$ entonces tenemos que $(1 + x^2 + x^3) \bmod g(x) = 1 + x^2 + \left(x^3 \bmod g(x)\right)$.
- Asi, $1 + x^2 + \left(x^3 \bmod g(x)\right) = 0$

Segundo método de codificación: Ejemplo

- Tomemos como antes $n = 7$, polinomio generador $g(x) = 1 + x^2 + x^3$ y $u(x) = 0110 = x + x^2$
- $u(x)x^{n-k} = (x + x^2)x^3 = x^4 + x^5$
- Debemos calcular $(x^4 + x^5) \bmod g(x)$.
- En principio debemos dividir $x^4 + x^5$ por $g(x)$ y obtener el resto, pero hay una forma mas fácil.
- Ciertamente $g(x) \bmod g(x) = 0$.
- Es decir $(1 + x^2 + x^3) \bmod g(x) = 0$.
- Por otro lado, como $gr(1 + x^2) < gr(g)$ entonces tenemos que $(1 + x^2 + x^3) \bmod g(x) = 1 + x^2 + \left(x^3 \bmod g(x)\right)$.
- Asi, $1 + x^2 + \left(x^3 \bmod g(x)\right) = 0$
- Por lo tanto $x^3 \bmod g(x) = 1 + x^2$.

Segundo método de codificación: Ejemplo

- Como $x^3 \bmod g(x) = 1 + x^2$ entonces multiplicando por x tenemos:

Segundo método de codificación: Ejemplo

- Como $x^3 \bmod g(x) = 1 + x^2$ entonces multiplicando por x tenemos:
- $x^4 \bmod g(x) = x(1 + x^2) \bmod g(x) = (x + x^3) \bmod g(x)$

Segundo método de codificación: Ejemplo

- Como $x^3 \bmod g(x) = 1 + x^2$ entonces multiplicando por x tenemos:
- $x^4 \bmod g(x) = x(1 + x^2) \bmod g(x) = (x + x^3) \bmod g(x)$
- Volviendo a usar que $x^3 \bmod g(x) = 1 + x^2$ obtenemos

Segundo método de codificación: Ejemplo

- Como $x^3 \bmod g(x) = 1 + x^2$ entonces multiplicando por x tenemos:
- $x^4 \bmod g(x) = x(1 + x^2) \bmod g(x) = (x + x^3) \bmod g(x)$
- Volviendo a usar que $x^3 \bmod g(x) = 1 + x^2$ obtenemos
- $x^4 \bmod g(x) = x + (1 + x^2) = 1 + x + x^2$.

Segundo método de codificación: Ejemplo

- Como $x^3 \bmod g(x) = 1 + x^2$ entonces multiplicando por x tenemos:
- $x^4 \bmod g(x) = x(1 + x^2) \bmod g(x) = (x + x^3) \bmod g(x)$
- Volviendo a usar que $x^3 \bmod g(x) = 1 + x^2$ obtenemos
- $x^4 \bmod g(x) = x + (1 + x^2) = 1 + x + x^2$.
- Y volviendo a multiplicar por x:

Segundo método de codificación: Ejemplo

- Como $x^3 \bmod g(x) = 1 + x^2$ entonces multiplicando por x tenemos:
- $x^4 \bmod g(x) = x(1 + x^2) \bmod g(x) = (x + x^3) \bmod g(x)$
- Volviendo a usar que $x^3 \bmod g(x) = 1 + x^2$ obtenemos
- $x^4 \bmod g(x) = x + (1 + x^2) = 1 + x + x^2$.
- Y volviendo a multiplicar por x:
- $x^5 \bmod g(x) = x + x^2 + x^3 \bmod g(x) = x + x^2 + 1 + x^2 = 1 + x$.

Segundo método de codificación: Ejemplo

- Como $x^3 \bmod g(x) = 1 + x^2$ entonces multiplicando por x tenemos:
- $x^4 \bmod g(x) = x(1 + x^2) \bmod g(x) = (x + x^3) \bmod g(x)$
- Volviendo a usar que $x^3 \bmod g(x) = 1 + x^2$ obtenemos
- $x^4 \bmod g(x) = x + (1 + x^2) = 1 + x + x^2$.
- Y volviendo a multiplicar por x:
- $x^5 \bmod g(x) = x + x^2 + x^3 \bmod g(x) = x + x^2 + 1 + x^2 = 1 + x$.
- Por lo tanto $(x^4 + x^5) \bmod g(x) = 1 + x + x^2 + 1 + x = x^2$.

Segundo método de codificación: Ejemplo

- Entonces $u(x) = x + x^2$ se codifica como:

Segundo método de codificación: Ejemplo

- Entonces $u(x) = x + x^2$ se codifica como:
- $(x^4 + x^5) \bmod g(x) + (x^4 + x^5) = x^2 + x^4 + x^5$.

Segundo método de codificación: Ejemplo

- Entonces $u(x) = x + x^2$ se codifica como:
- $(x^4 + x^5) \bmod g(x) + (x^4 + x^5) = x^2 + x^4 + x^5$.
- Es decir, la palabra 0110 como la palabra 0010110

Segundo método de codificación: Ejemplo

- Entonces $u(x) = x + x^2$ se codifica como:
- $(x^4 + x^5) \bmod g(x) + (x^4 + x^5) = x^2 + x^4 + x^5$.
- Es decir, la palabra 0110 como la palabra 0010110
- Oberven que 0110 “está" en 0010110

Segundo método de codificación: Ejemplo

- Entonces $u(x) = x + x^2$ se codifica como:
- $(x^4 + x^5) \bmod g(x) + (x^4 + x^5) = x^2 + x^4 + x^5$.
- Es decir, la palabra 0110 como la palabra 0010110
- Oberven que 0110 “está" en 0010110

Segundo método de codificación: Ejemplo

- Entonces $u(x) = x + x^2$ se codifica como:
- $(x^4 + x^5) \bmod g(x) + (x^4 + x^5) = x^2 + x^4 + x^5$.
- Es decir, la palabra 0110 como la palabra 0010110
- Oberven que 0110 “está" en 0010110
- Que es lo que habiamos explicado antes.

Segundo método de codificación: Ejemplo

- Entonces $u(x) = x + x^2$ se codifica como:
- $(x^4 + x^5) \bmod g(x) + (x^4 + x^5) = x^2 + x^4 + x^5$.
- Es decir, la palabra 0110 como la palabra 0010110
- Oberven que 0110 “está" en 0010110
- Que es lo que habiamos explicado antes.
- Asi que de 0010110 es fácil recuperar u: basta mirar los últimos 4 bits.

Segundo método de codificación: Ejemplo

- Entonces $u(x) = x + x^2$ se codifica como:
- $(x^4 + x^5) \bmod g(x) + (x^4 + x^5) = x^2 + x^4 + x^5$.
- Es decir, la palabra 0110 como la palabra 0010110
- Oberven que 0110 “está" en 0010110
- Que es lo que habiamos explicado antes.
- Asi que de 0010110 es fácil recuperar u: basta mirar los últimos 4 bits.
- En general,hay que mirar los últimos k bits, por la explicación que habiamos dado antes.

- Todo esto parece mucho calculo para codificar una palabra, y lo es.

Segundo método de codificación

- Todo esto parece mucho calculo para codificar una palabra, y lo es.
- Pero uno no codifica UNA palabra.

Segundo método de codificación

- Todo esto parece mucho calculo para codificar una palabra, y lo es.
- Pero uno no codifica UNA palabra.
- Todos esos calculos sirven para todas las otras palabras.

Segundo método de codificación

- Todo esto parece mucho calculo para codificar una palabra, y lo es.
- Pero uno no codifica UNA palabra.
- Todos esos calculos sirven para todas las otras palabras.
- (en realidad, todavia nos faltaria calcular $x^6 \bmod g(x)$).

Segundo método de codificación

- Todo esto parece mucho calculo para codificar una palabra, y lo es.
- Pero uno no codifica UNA palabra.
- Todos esos calculos sirven para todas las otras palabras.
- (en realidad, todavia nos faltaria calcular $x^6 \bmod g(x)$).
- Por ejemplo si queremos codificar $1010 = 1 + x^2$, la codificación seria:

Segundo método de codificación

- Todo esto parece mucho calculo para codificar una palabra, y lo es.
- Pero uno no codifica UNA palabra.
- Todos esos calculos sirven para todas las otras palabras.
- (en realidad, todavia nos faltaria calcular $x^6 \bmod g(x)$).
- Por ejemplo si queremos codificar $1010 = 1 + x^2$, la codificación seria:

$$(1+x^2)x^3 \bmod g(x) + (1+x^2)x^3 \quad = \quad (x^3+x^5) \bmod g(x) + x^3 + x^5$$

Segundo método de codificación

- Todo esto parece mucho calculo para codificar una palabra, y lo es.
- Pero uno no codifica UNA palabra.
- Todos esos calculos sirven para todas las otras palabras.
- (en realidad, todavia nos faltaria calcular $x^6 \bmod g(x)$).
- Por ejemplo si queremos codificar $1010 = 1 + x^2$, la codificación seria:

$$\begin{aligned}(1+x^2)x^3 \bmod g(x) + (1+x^2)x^3 &= (x^3+x^5) \bmod g(x) + x^3 + x^5 \\ &= 1 + x^2 + 1 + x + x^3 + x^5\end{aligned}$$

Segundo método de codificación

- Todo esto parece mucho calculo para codificar una palabra, y lo es.
- Pero uno no codifica UNA palabra.
- Todos esos calculos sirven para todas las otras palabras.
- (en realidad, todavia nos faltaria calcular $x^6 \bmod g(x)$).
- Por ejemplo si queremos codificar $1010 = 1 + x^2$, la codificación seria:

$$\begin{array}{rcl} (1+x^2)x^3 \bmod g(x) + (1+x^2)x^3 & = & (x^3+x^5) \bmod g(x) + x^3 + x^5 \\ \text{(de los calculos que hicimos antes)} & = & 1 + x^2 + 1 + x + x^3 + x^5 \end{array}$$

Segundo método de codificación

- Todo esto parece mucho calculo para codificar una palabra, y lo es.
- Pero uno no codifica UNA palabra.
- Todos esos calculos sirven para todas las otras palabras.
- (en realidad, todavia nos faltaria calcular $x^6 \bmod g(x)$).
- Por ejemplo si queremos codificar $1010 = 1 + x^2$, la codificación seria:

$$\begin{aligned}(1+x^2)x^3 \bmod g(x) + (1+x^2)x^3 &= (x^3+x^5) \bmod g(x) + x^3 + x^5 \\ &= 1 + x^2 + 1 + x + x^3 + x^5 \\ &= x + x^2 + x^3 + x^5\end{aligned}$$

Segundo método de codificación

- Asi que 1010 se codifica como 0111010.

Segundo método de codificación

- Asi que 1010 se codifica como 0111010.
- Un ejemplo mas: 1101.

Segundo método de codificación

- Asi que 1010 se codifica como 0111010.
- Un ejemplo mas: 1101.
- Tenemos $1 + x + x^3 \mapsto (x^3 + x^4 + x^6) \bmod g(x) + x^3 + x^4 + x^6$

Segundo método de codificación

- Asi que 1010 se codifica como 0111010.
- Un ejemplo mas: 1101.
- Tenemos $1 + x + x^3 \mapsto (x^3 + x^4 + x^6) \bmod g(x) + x^3 + x^4 + x^6$
- Vamos a necesitar $x^6 \mod g(x)$.

Segundo método de codificación

- Asi que 1010 se codifica como 0111010.
- Un ejemplo mas: 1101.
- Tenemos $1 + x + x^3 \mapsto (x^3 + x^4 + x^6) \bmod g(x) + x^3 + x^4 + x^6$
- Vamos a necesitar $x^6 \mod g(x)$.
- Lo sacamos multiplicando por x a $x^5 \bmod g(x) = 1 + x$:

- Asi que 1010 se codifica como 0111010.
- Un ejemplo mas: 1101.
- Tenemos $1 + x + x^3 \mapsto (x^3 + x^4 + x^6) \bmod g(x) + x^3 + x^4 + x^6$
- Vamos a necesitar $x^6 \mod g(x)$.
- Lo sacamos multiplicando por x a $x^5 \bmod g(x) = 1 + x$:
  - $x^6 \bmod g(x) = x + x^2$

Segundo método de codificación

- Asi que 1010 se codifica como 0111010.
- Un ejemplo mas: 1101.
- Tenemos $1 + x + x^3 \mapsto (x^3 + x^4 + x^6) \bmod g(x) + x^3 + x^4 + x^6$
- Vamos a necesitar $x^6 \mod g(x)$.
- Lo sacamos multiplicando por x a $x^5 \bmod g(x) = 1 + x$:
  - $x^6 \bmod g(x) = x + x^2$
- Por lo tanto la codificación es:

$$x^3 \mod g(x) + x^4 \mod g(x) + x^6 \mod g(x) + x^3 + x^4 + x^6$$

Segundo método de codificación

- Asi que 1010 se codifica como 0111010.
- Un ejemplo mas: 1101.
- Tenemos $1 + x + x^3 \mapsto (x^3 + x^4 + x^6) \bmod g(x) + x^3 + x^4 + x^6$
- Vamos a necesitar $x^6 \mod g(x)$.
- Lo sacamos multiplicando por x a $x^5 \bmod g(x) = 1 + x$:
  - $x^6 \bmod g(x) = x + x^2$
- Por lo tanto la codificación es:

$$(1 + x^2) + x^4 \mod g(x) + x^6 \mod g(x) + x^3 + x^4 + x^6$$

Segundo método de codificación

- Asi que 1010 se codifica como 0111010.
- Un ejemplo mas: 1101.
- Tenemos $1 + x + x^3 \mapsto (x^3 + x^4 + x^6) \bmod g(x) + x^3 + x^4 + x^6$
- Vamos a necesitar $x^6 \mod g(x)$.
- Lo sacamos multiplicando por x a $x^5 \bmod g(x) = 1 + x$:
  - $x^6 \bmod g(x) = x + x^2$
- Por lo tanto la codificación es:

$$(1 + x^2) + (1 + x + x^2) + x^6 \mod g(x) + x^3 + x^4 + x^6$$

Segundo método de codificación

- Asi que 1010 se codifica como 0111010.
- Un ejemplo mas: 1101.
- Tenemos $1 + x + x^3 \mapsto (x^3 + x^4 + x^6) \bmod g(x) + x^3 + x^4 + x^6$
- Vamos a necesitar $x^6 \mod g(x)$.
- Lo sacamos multiplicando por x a $x^5 \bmod g(x) = 1 + x$:
  - $x^6 \bmod g(x) = x + x^2$
- Por lo tanto la codificación es:

$$(1 + x^2) + (1 + x + x^2) + (x + x^2) + x^3 + x^4 + x^6$$

Segundo método de codificación

- Asi que 1010 se codifica como 0111010.
- Un ejemplo mas: 1101.
- Tenemos $1 + x + x^3 \mapsto (x^3 + x^4 + x^6) \bmod g(x) + x^3 + x^4 + x^6$
- Vamos a necesitar $x^6 \mod g(x)$.
- Lo sacamos multiplicando por x a $x^5 \bmod g(x) = 1 + x$:
  - $x^6 \bmod g(x) = x + x^2$
- Por lo tanto la codificación es:

$$x^2 + x^3 + x^4 + x^6$$

Segundo método de codificación

- Asi que 1010 se codifica como 0111010.
- Un ejemplo mas: 1101.
- Tenemos $1 + x + x^3 \mapsto (x^3 + x^4 + x^6) \bmod g(x) + x^3 + x^4 + x^6$
- Vamos a necesitar $x^6 \mod g(x)$.
- Lo sacamos multiplicando por x a $x^5 \bmod g(x) = 1 + x$:
  - $x^6 \bmod g(x) = x + x^2$
- Por lo tanto la codificación es:

$$x^2 + x^3 + x^4 + x^6 = 0011101$$

Chequeando que $1 + x^n$ sea divisible por $g(x)$

- El teorema dice que $g(x)$ divide a $1 + x^n$.

Chequeando que $1 + x^n$ sea divisible por $g(x)$

- El teorema dice que $g(x)$ divide a $1 + x^n$.
- Les podemos pedir que verifiquen esto.

Chequeando que $1 + x^n$ sea divisible por $g(x)$

- El teorema dice que $g(x)$ divide a $1 + x^n$.
- Les podemos pedir que verifiquen esto.
- La idea no es que dividan.

Chequeando que $1 + x^n$ sea divisible por $g(x)$

- El teorema dice que $g(x)$ divide a $1 + x^n$.
- Les podemos pedir que verifiquen esto.
- La idea no es que dividan.
- En nuestro ejemplo, a partir de $x^6 \bmod g(x) = x + x^2$, multiplicamos por x y obtenemos:

Chequeando que $1 + x^n$ sea divisible por $g(x)$

- El teorema dice que $g(x)$ divide a $1 + x^n$.
- Les podemos pedir que verifiquen esto.
- La idea no es que dividan.
- En nuestro ejemplo, a partir de $x^6 \bmod g(x) = x + x^2$, multiplicamos por x y obtenemos:
- $x^7 \bmod g(x) = x^2 + x^3 \bmod g(x) = x^2 + 1 + x^2 = 1$

Chequeando que $1+x^n$ sea divisible por $g(x)$

- El teorema dice que $g(x)$ divide a $1+x^n$.
- Les podemos pedir que verifiquen esto.
- La idea no es que dividan.
- En nuestro ejemplo, a partir de $x^6 \bmod g(x) = x + x^2$, multiplicamos por x y obtenemos:
- $x^7 \bmod g(x) = x^2 + x^3 \bmod g(x) = x^2 + 1 + x^2 = 1$
- Lo cual dice que $1 + x^7 \bmod g(x) = 0$.

Chequeando que $1 + x^n$ sea divisible por $g(x)$

- El teorema dice que $g(x)$ divide a $1 + x^n$.
- Les podemos pedir que verifiquen esto.
- La idea no es que dividan.
- En nuestro ejemplo, a partir de $x^6 \bmod g(x) = x + x^2$, multiplicamos por x y obtenemos:
- $x^7 \bmod g(x) = x^2 + x^3 \bmod g(x) = x^2 + 1 + x^2 = 1$
- Lo cual dice que $1 + x^7 \bmod g(x) = 0$.
- Esto sirve para chequear que no se hayan equivocado en alguna cuenta al hacer todas las congruencias

Matriz generadora para el segundo método de codificación

- Una matriz generadora va a venir dada por la codificación de $1, x, x^2, ..., x^{k-1}$

Matriz generadora para el segundo método de codificación

- Una matriz generadora va a venir dada por la codificación de $1, x, x^2, ..., x^{k-1}$
- Es decir, la matriz:

$$\begin{bmatrix} x^{n-k} \mod g(x) + x^{n-k} \\ x^{n-k+1} \mod g(x) + x^{n-k+1} \\ x^{n-k+2} \mod g(x) + x^{n-k+2} \\ x^{n-k+3} \mod g(x) + x^{n-k+3} \\ \dots \\ \dots \\ x^{n-1} \mod g(x) + x^{n-1} \end{bmatrix}$$

Matriz generadora para el segundo método de codificación

- En nuestro ejemplo seria:

Matriz generadora para el segundo método de codificación

- En nuestro ejemplo seria:

$$\begin{bmatrix} x^3 \mod g(x) & + & x^3 \\ x^4 \mod g(x) & + & x^4 \\ x^5 \mod g(x) & + & x^5 \\ x^6 \mod g(x) & + & x^6 \end{bmatrix}$$

Matriz generadora para el segundo método de codificación

- En nuestro ejemplo seria:

$$\begin{bmatrix} 1+x^2 & + & x^3 \\ 1+x+x^2 & + & x^4 \\ 1+x & + & x^5 \\ x+x^2 & + & x^6 \end{bmatrix}$$

Matriz generadora para el segundo método de codificación

- En nuestro ejemplo seria:

$$\begin{bmatrix} 1+x^2 & + & x^3 \\ 1+x+x^2 & + & x^4 \\ 1+x & + & x^5 \\ x+x^2 & + & x^6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz generadora para el segundo método de codificación

- En nuestro ejemplo seria:

$$\begin{bmatrix} 1+x^2 & + & x^3 \\ 1+x+x^2 & + & x^4 \\ 1+x & + & x^5 \\ x+x^2 & + & x^6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Observemos que tiene la identidad a derecha, como tiene que ser de toda la discusión que hemos venido haciendo

Matriz de chequeo

- Como esta matriz generadora es de la forma $[A|I_4]$, entonces una matriz de chequeo tendrá la forma $[I_3|A^t]$:

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

Matriz de chequeo

- Como esta matriz generadora es de la forma $[A|I_4]$, entonces una matriz de chequeo tendrá la forma $[I_3|A^t]$:

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

- Esta es la matriz de un código de Hamming.

Matriz de chequeo

- Como esta matriz generadora es de la forma $[A|I_4]$, entonces una matriz de chequeo tendrá la forma $[I_3|A^t]$:

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

- Esta es la matriz de un código de Hamming.
- Se puede ver que todos los códigos de Hamming son (en algun orden de las columnas) códigos cíclicos.

Matriz de chequeo

- Como esta matriz generadora es de la forma $[A|I_4]$, entonces una matriz de chequeo tendrá la forma $[I_3|A^t]$:

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

- Esta es la matriz de un código de Hamming.
- Se puede ver que todos los códigos de Hamming son (en algun orden de las columnas) códigos cíclicos.
- La matriz de chequeo con la identidad a izquierda se puede obtener directamente sin pasar por la generadora pues la columna j-ésima es $x^j \bmod g(x)$, claramente de toda la discusión que hemos hecho. (ver la matriz de arriba)

Otro ejemplo

- Veamos otro ejemplo: $g(x) = 1 + x^2 + x^3 + x^4$, $n = 7$.

Otro ejemplo

- Veamos otro ejemplo: $g(x) = 1 + x^2 + x^3 + x^4$, $n = 7$.
- $k = 7 - 4 = 3$.

Otro ejemplo

- Veamos otro ejemplo: $g(x) = 1 + x^2 + x^3 + x^4$, $n = 7$.
- $k = 7 - 4 = 3$.
- $x^4 \bmod g(x) = 1 + x^2 + x^3$.

Otro ejemplo

- Veamos otro ejemplo: $g(x) = 1 + x^2 + x^3 + x^4$, $n = 7$.
- $k = 7 - 4 = 3$.
- $x^4 \bmod g(x) = 1 + x^2 + x^3$.
- $x^5 \bmod g(x) = x + x^3 + x^4 \bmod g(x)$

Otro ejemplo

- Veamos otro ejemplo: $g(x) = 1 + x^2 + x^3 + x^4$, $n = 7$.
- $k = 7 - 4 = 3$.
- $x^4 \bmod g(x) = 1 + x^2 + x^3$.
- $x^5 \bmod g(x) = x + x^3 + x^4 \bmod g(x)$
- Usando $x^4 \bmod g(x) = 1 + x^2 + x^3$:

Otro ejemplo

- Veamos otro ejemplo: $g(x) = 1 + x^2 + x^3 + x^4$, $n = 7$.
- $k = 7 - 4 = 3$.
- $x^4 \bmod g(x) = 1 + x^2 + x^3$.
- $x^5 \bmod g(x) = x + x^3 + x^4 \bmod g(x)$
- Usando $x^4 \bmod g(x) = 1 + x^2 + x^3$:
- $x^5 \bmod g(x) = x + x^3 + 1 + x^2 + x^3 = 1 + x + x^2$.

Otro ejemplo

- Veamos otro ejemplo: $g(x) = 1 + x^2 + x^3 + x^4$, $n = 7$.
- $k = 7 - 4 = 3$.
- $x^4 \bmod g(x) = 1 + x^2 + x^3$.
- $x^5 \bmod g(x) = x + x^3 + x^4 \bmod g(x)$
- Usando $x^4 \bmod g(x) = 1 + x^2 + x^3$:
- $x^5 \bmod g(x) = x + x^3 + 1 + x^2 + x^3 = 1 + x + x^2$.
- $x^6 \bmod g(x) = x + x^2 + x^3$

Otro ejemplo

- Veamos otro ejemplo: $g(x) = 1 + x^2 + x^3 + x^4$, $n = 7$.
- $k = 7 - 4 = 3$.
- $x^4 \bmod g(x) = 1 + x^2 + x^3$.
- $x^5 \bmod g(x) = x + x^3 + x^4 \bmod g(x)$
- Usando $x^4 \bmod g(x) = 1 + x^2 + x^3$:
- $x^5 \bmod g(x) = x + x^3 + 1 + x^2 + x^3 = 1 + x + x^2$.
- $x^6 \bmod g(x) = x + x^2 + x^3$
- Por lo tanto, la matrix generadora con la identidad a derecha es la de la siguiente pagina

Otro ejemplo

- Veamos otro ejemplo: $g(x) = 1 + x^2 + x^3 + x^4$, $n = 7$.
- $k = 7 - 4 = 3$.
- $x^4 \bmod g(x) = 1 + x^2 + x^3$.
- $x^5 \bmod g(x) = x + x^3 + x^4 \bmod g(x)$
- Usando $x^4 \bmod g(x) = 1 + x^2 + x^3$:
- $x^5 \bmod g(x) = x + x^3 + 1 + x^2 + x^3 = 1 + x + x^2$.
- $x^6 \bmod g(x) = x + x^2 + x^3$
- Por lo tanto, la matrix generadora con la identidad a derecha es la de la siguiente pagina
- Pero antes hagamos el Check:

Otro ejemplo

- Veamos otro ejemplo: $g(x) = 1 + x^2 + x^3 + x^4$, $n = 7$.
- $k = 7 - 4 = 3$.
- $x^4 \bmod g(x) = 1 + x^2 + x^3$.
- $x^5 \bmod g(x) = x + x^3 + x^4 \bmod g(x)$
- Usando $x^4 \bmod g(x) = 1 + x^2 + x^3$:
- $x^5 \bmod g(x) = x + x^3 + 1 + x^2 + x^3 = 1 + x + x^2$.
- $x^6 \bmod g(x) = x + x^2 + x^3$
- Por lo tanto, la matrix generadora con la identidad a derecha es la de la siguiente pagina
- Pero antes hagamos el Check:
- $x^7 \bmod g(x) = x^2 + x^3 + x^4 \bmod g(x)$

Otro ejemplo

- Veamos otro ejemplo: $g(x) = 1 + x^2 + x^3 + x^4$, $n = 7$.
- $k = 7 - 4 = 3$.
- $x^4 \bmod g(x) = 1 + x^2 + x^3$.
- $x^5 \bmod g(x) = x + x^3 + x^4 \bmod g(x)$
- Usando $x^4 \bmod g(x) = 1 + x^2 + x^3$:
- $x^5 \bmod g(x) = x + x^3 + 1 + x^2 + x^3 = 1 + x + x^2$.
- $x^6 \bmod g(x) = x + x^2 + x^3$
- Por lo tanto, la matrix generadora con la identidad a derecha es la de la siguiente pagina
- Pero antes hagamos el Check:
- $x^7 \bmod g(x) = x^2 + x^3 + 1 + x^2 + x^3$

Otro ejemplo

- Veamos otro ejemplo: $g(x) = 1 + x^2 + x^3 + x^4$, $n = 7$.
- $k = 7 - 4 = 3$.
- $x^4 \bmod g(x) = 1 + x^2 + x^3$.
- $x^5 \bmod g(x) = x + x^3 + x^4 \bmod g(x)$
- Usando $x^4 \bmod g(x) = 1 + x^2 + x^3$:
- $x^5 \bmod g(x) = x + x^3 + 1 + x^2 + x^3 = 1 + x + x^2$.
- $x^6 \bmod g(x) = x + x^2 + x^3$
- Por lo tanto, la matrix generadora con la identidad a derecha es la de la siguiente pagina
- Pero antes hagamos el Check:
- $x^7 \bmod g(x) = x^2 + x^3 + 1 + x^2 + x^3 = 1$

Ejemplo

$$\begin{bmatrix} x^4 \mod g(x) & + & x^4 \\ x^5 \mod g(x) & + & x^5 \\ x^6 \mod g(x) & + & x^6 \end{bmatrix}$$

Ejemplo

$$\begin{bmatrix} 1+x^2+x^3 & + & x^4 \\ 1+x+x^2 & + & x^5 \\ x+x^2+x^3 & + & x^6 \end{bmatrix}$$

Ejemplo

$$\begin{bmatrix} 1+x^2+x^3 & + & x^4 \\ 1+x+x^2 & + & x^5 \\ x+x^2+x^3 & + & x^6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Ejemplo

$$\begin{bmatrix} 1+x^2+x^3 & + & x^4 \\ 1+x+x^2 & + & x^5 \\ x+x^2+x^3 & + & x^6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Con matriz de chequeo:

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

Error Trapping

- Algunos códigos cíclicos muy usados tiene la propiedad de corregir mas de un error.

Error Trapping

- Algunos códigos cíclicos muy usados tiene la propiedad de corregir mas de un error.
- Para corregir esos errores, hay un algoritmo llamado “error trapping” que permite corregir esos errores en la mayoria de los casos (no siempre) y que es fácil de implementar en hardware.

Error Trapping

- Algunos códigos cíclicos muy usados tiene la propiedad de corregir mas de un error.
- Para corregir esos errores, hay un algoritmo llamado “error trapping” que permite corregir esos errores en la mayoria de los casos (no siempre) y que es fácil de implementar en hardware.
- Otros años lo hemos dado pero este año no lo daremos.