Códigos Cíclicos, 2da Clase

Daniel Penazzi

May 28, 2022

Tabla de Contenidos

Teorema fundamental de códigos cíclicos

La clase pasada vieron con Diego el Teorema fundamental de códigos cíclicos, que era el siguiente:

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. $C = \{v(x) \odot g(x) : v$ es un polinomio cualquiera$\}$
3. $gr(g(x)) = n - k$.
4. $g(x)$ divide a $1 + x^n$
5. $g_0 = 1$

Primer método de codificación

- El teorema fundamental de códigos cíclicos 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 $gr(u(x)g(x)) = gr(u) + gr(g) < k + n - k = n$).

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
- 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 $\{0,1\}^k$.
- 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.
- 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.
- 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 $(u(x) \bmod g(x)) + u(x)$"
- 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:
- $(u(x) \bmod g(x)) + u(x) = u(x) + u(x) = 0$ para todo $u(x)$ de grado menor que 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: $(u(x) + w(x)))\, x^{n-k} = (u(x) + w(x))x^{n-k} \bmod g(x))$.
- 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.
- 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 $u(x) = 0110 = x + x^2$
- $u(x)x^{n-k} = (x + x^2)x^3 = x^4 + x^5$
- 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:
- $x^4 \bmod g(x) = x(1 + x^2) \bmod g(x) = (x + x^3) \bmod g(x)$
- 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:
- $(x^4 + x^5) \bmod g(x) + (x^4 + x^5) = x^2 + x^4 + x^5$.
- 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.

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.
- 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 $1 + x^n$.
- 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 $1, x, x^2, ..., x^{k-1}$
- 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:

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

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 $[I_3|A^t]$:

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

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

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

Con matriz de chequeo:

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

Polinomio chequeador

- 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
- $h(x)p(x) \bmod (1 + x^n) = 0$, 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
- 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 dimos.