

Definiciones

miércoles, 27 de abril de 2022 15:12

Se usan todas las definiciones de álgebra lineal

X^n

En esta materia se define X^n como el vector horizontal

Por ejemplo:

$$[x_1 \ x_2 \ \cdots \ x_n] \in X^n$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \notin X^n$$

Esto implica que:

$$(x_1, x_2, \dots, x_n) = [x_1 \ x_2 \ \cdots \ x_n]$$

(porque $(, \dots,)$ representa elementos de X^n)

$X^{x \times y}$

Son matrices con x filas e y columnas

Código:

Sea A un conjunto

$$n \in \mathbb{N}$$

Definiciones genéricas:

C es un código sobre $A \Leftrightarrow C \subseteq A^*$

C es un código de bloque sobre $A \Leftrightarrow \exists k \in \mathbb{N} : C \subseteq A^k$

C es un código de longitud n sobre $A \Leftrightarrow C \subseteq A^n$

Como voy a trabajar siempre con códigos binarios de bloque, voy a usar código para código de bloque sobre $\{0,1\}$, es decir:

C es un código $\coloneqq C$ es un código de bloque sobre $\{0,1\}$

Distancia:

Sea:

C un código de bloque de longitud n sobre A

$$v, w \in C$$

$$d(v, w) = \#\{i \in \mathbb{N}_{\leq n} : v_i \neq w_i\}$$

$$\delta(C) = \min\{d(s, u) : s, u \in C : s \neq u\}$$

Disco de radio r :

Sea:

$$v \in A^n$$

$$r \in \mathbb{N}_0$$

$$D_r(v) = \{w \in A : d(v, w) \leq r\}$$

Detección y corrección de errores:

Sea:

C un código de bloque de longitud n sobre A

$$r \in \mathbb{N}_0$$

C detecta r errores $\Leftrightarrow \forall v \in C : D_r(v) \cap C = \{v\}$

C corrige r errores $\Leftrightarrow \forall v, w \in C : D_r(v) \cap D_r(w) = \emptyset$

Código perfecto:

Sea C un código de longitud n

$$C \text{ es perfecto} \Leftrightarrow \#C = \frac{2^n}{\sum_{r=0}^{\lfloor \frac{\delta(C)-1}{2} \rfloor} \binom{n}{r}}$$

Código lineal:

Sea $n \in \mathbb{N}$

C es un código lineal de longitud $n \Leftrightarrow C$ es un subespacio vectorial de $\{0,1\}^n$

Peso de Hamming:

Sea:

$$v \in \{0,1\}^n$$

$$|v| = d(v, 0)$$

Parámetros del código:

Sea:

C un código lineal

C tiene parámetros (n, k, δ)

$$\Leftrightarrow C \text{ es de longitud } n \wedge \dim(C) = k \wedge \delta(C) = \delta$$

Sea:

$$k, n \in \mathbb{N}$$

$G \in \{0,1\}^{k \times n}$ cuyas filas son LI

El código generado por G es espacioFila(G)

Matriz de chequeo:

Sea:

$$n, r \in \mathbb{N}$$

C un código lineal de longitud n

$$H \in \{0,1\}^{r \times n}$$

H es matriz de chequeo de $C \Leftrightarrow \text{Nu}(T) = C$

Donde: $T(x) = Hx^t$

Código de Hamming:

Sea:

$$r \in \mathbb{N}$$

$$\mathcal{H}_r = \{C \text{ códigos lineales : } C \text{ tiene parámetros } (2^r - 1, 2^r - r - 1, 3)\}$$

Teoremas

Lunes, 2 de mayo de 2022 21:43

d es una distancia:

Sea

C un código

$v, w, u \in C$

$$d(v, w) = d(w, v)$$

$$d(v, w) \geq 0$$

$$d(v, w) = 0 \Leftrightarrow v = w$$

$$d(v, w) \leq d(v, u) + d(u, w)$$

Teoremas de δ :

Sea:

Sea C un código de longitud n

C detecta $\delta(C) - 1$ errores

C no detecta $\delta(C)$ errores

C corrige $\left\lfloor \frac{\delta(C) - 1}{2} \right\rfloor$ errores

C no corrige $\left\lfloor \frac{\delta(C) - 1}{2} \right\rfloor + 1$ errores

$$\#C \leq \frac{2^n}{\sum_{r=0}^{\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor} \binom{n}{r}}$$

Teoremas asociados al peso de Hamming:

Sean:

$v, w \in \{0,1\}^n$

C un código lineal

$$|v| = \sum_{i=1}^n v_i$$

$$d(v, w) = |v + w|$$

$$\delta(C) = \min\{|u| : u \in C - \{0\}\}$$

Cantidad de elementos de un código:

Sea:

C un código lineal de longitud n

$k = \dim(C)$

G una matriz generadora de C

$$\#C = 2^k$$

$$G \in \{0,1\}^{k \times n}$$

Los códigos de Hamming son los mejores de $\delta = 3$:

Sea:

$$r \in \mathbb{N}$$

$\exists C$ código lineal de longitud $2^r - 1 : \dim(C) > 2^r - r - 1 \wedge \delta(C) = 3$

Matriz de chequeo:

Sean:

$$n, k \in \mathbb{N}$$

C un código lineal

$$A, B \in \{0,1\}^{k \times n-k}$$

$[\text{Id}_k \quad A], [B \quad \text{Id}_k]$ generan a C

$[A^t \quad \text{Id}_{n-k}], [\text{Id}_{n-k} \quad B^t]$ son matrices de chequeo de C

Sean:

C un código lineal de longitud n

H una matriz de chequeo de C

$H \in \{0,1\}^{r \times n}$

$$\delta(C) = \min\{\#D : D \subseteq \text{columnas}(H) \wedge D \text{ es LD}\}$$

Las filas de H son LI $\Leftrightarrow r = n - \dim(C)$

Corrección de errores:

Sea:

C un código lineal de longitud n

H una matriz de chequeo de C de tamaño $r \times n$

$\alpha \in \{0,1\}^n$

$$H\alpha^t = 0 \Rightarrow \alpha \in C$$

$$H\alpha^t = \text{columna}_i(H) \Rightarrow \underset{\beta \in C - \{\alpha\}}{\min \arg} d(\alpha, \beta) = \alpha + e_i$$

$$H\alpha^t = \sum_{j=0}^k \text{columna}_{ij}(H) \Rightarrow \underset{\beta \in C - \{\alpha\}}{\min \arg} d(\alpha, \beta) = \alpha + \sum_{j=0}^k e_{ij}$$

($\underset{\beta \in C - \{\alpha\}}{\min \arg} d(\alpha, \beta)$ lese como "palabra de C mas cercana a α ")

Cota de Singleton:

Sea:

C un código lineal de longitud n

$$\delta(C) \leq n - \dim(C) + 1$$

Demostraciones

miércoles, 4 de mayo de 2022 11:57

Teorema de δ :

Sea C un código

C detecta $\delta(C) - 1$ errores

C no detecta $\delta(C)$ errores

C corrige $\left\lfloor \frac{\delta(C) - 1}{2} \right\rfloor$ errores

C no corrige $\left\lceil \frac{\delta(C) - 1}{2} \right\rceil + 1$ errores

Demostraciones:

Sea n el tamaño de C

C detecta $\delta(C) - 1$ errores:

C detecta $\delta(C) - 1$ errores

$\Leftrightarrow \{\text{Definición detectar errores}\}$

$\forall v \in C : D_{\delta(C)-1}(v) \cap C = \{v\}$

Trabajo sin el \forall :

$D_{\delta(C)-1}(v) \cap C = \{v\}$

$\Leftrightarrow \{\text{Definición } D_r\}$

$\{w \in \{0,1\}^n : d(v, w) \leq \delta(C) - 1\} \cap C = \{v\}$

\Leftrightarrow

$\{w \in C : d(v, w) \leq \delta(C) - 1\} = \{v\}$

\Leftrightarrow

$d(v, v) \leq \delta(C) - 1 \wedge \langle \exists w \in C - \{v\} : d(v, w) \leq \delta(C) - 1 \rangle$

\Leftrightarrow

$0 \leq \delta(C) - 1 \wedge \langle \exists w \in C - \{v\} : d(v, w) < \delta(C) \rangle$

$\Leftrightarrow \{\delta(C) \geq 1, \text{ definición } \delta\}$

$\exists w \in C - \{v\} : d(v, w) < \min\{d(s, u) : s, u \in C : s \neq u\}$

$\Leftrightarrow \{\text{Esto es claramente cierto}\}$

True

C no detecta $\delta(C)$ errores:

C no detecta $\delta(C)$ errores

$\Leftrightarrow \{\text{Definición detectar errores (negada)}\}$

$\exists v \in C : D_{\delta(C)}(v) \cap C \neq \{v\}$

$\Leftrightarrow \{\text{Definición } D_r\}$

$\exists v \in C : \{w \in C : d(v, w) \leq \delta(C)\} \neq \{v\}$

$\Leftrightarrow \{\text{Definición } \delta\}$

$\exists v \in C : \{w \in C : d(v, w) \leq \min\{d(s, u) : s, u \in C : s \neq u\}\} \neq \{v\}$

\Leftrightarrow

$\exists v, w \in C : v \neq w : d(v, w) \leq \min\{d(s, u) : s, u \in C : s \neq u\}$

$\Leftrightarrow \{\text{Existen, } v = s, w = u, \text{ con los } s, u \text{ que minimizan } d(s, u)\}$

True

C corrige $\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor$ errores:

C corrige $\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor$ errores

$\Leftrightarrow \{\text{Definición corregir errores}\}$

$$\forall v, w \in C : D_{\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor}(v) \cup D_{\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor}(w) = \emptyset$$

Trabajo sin el \forall :

$$D_{\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor}(v) \cup D_{\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor}(w) = \emptyset$$

$\Leftrightarrow \{\text{Definición } D_r\}$

$$\left\{ u \in \{0,1\}^n : d(v, u) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor \right\} \cup \left\{ u \in \{0,1\}^n : d(w, u) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor \right\} = \emptyset$$

\Leftrightarrow

$$\exists u \in \{0,1\}^n : d(v, u) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor \wedge d(w, u) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor$$

Demuestro esto por contradicción, suponiendo que si existen:

$$d(v, u) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor \wedge d(w, u) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor$$

$\Rightarrow \{\text{Desigualdad triangular}\}$

$$d(v, w) \leq 2 \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor$$

\Leftrightarrow

$$d(v, w) \leq 2 \frac{\delta(C)-1}{2}$$

\Leftrightarrow

$$d(v, w) \leq \delta(C) - 1$$

\Leftrightarrow

$$d(v, w) < \delta(C)$$

$\Leftrightarrow \{\delta(C) \text{ es la mínima distancia}\}$

False

C no corrige $\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1$ errores:

C no corrige $\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1$ errores

$\Leftrightarrow \{\text{Definición corregir errores (negada)}\}$

$$\exists v, w \in C : D_{\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1}(v) \cup D_{\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1}(w) \neq \emptyset$$

$\Leftrightarrow \{\text{Definición } D_r\}$

$$\exists v, w \in C : \left\{ u \in \{0,1\}^n : d(v, u) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1 \right\} \cup \left\{ u \in \{0,1\}^n : d(w, u) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1 \right\} \neq \emptyset$$

\Leftrightarrow

$$\exists v, w \in C, u \in \{0,1\}^n : d(v, u) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1 \wedge d(w, u) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1$$

Doy estos v, w, u :

Sea:

v, w talque $d(v, w) = \delta(C)$ (existen por definición de δ)

$$I = \{i \in \mathbb{N}_{\leq n} : v_i \neq w_i\}$$

$J \subseteq I$ talque $\#J = \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1$ (cualquier J que lo cumpla sirve)

$$u_i = \begin{cases} i \in J & \rightarrow 1 - v_i \\ \text{si no} & \rightarrow v_i \end{cases}$$

J existe porque $\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1 \leq \delta(C) = d(v, w) = \#I$

Pruebo que se satisfacen las condiciones:

$$d(v, u) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1$$

$\Leftrightarrow \{\text{Por construcción de } u\}$

$$\#J \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1$$

\Leftrightarrow

$$\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1 \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1$$

\Leftrightarrow

True

$$d(w, u) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1$$

$\Leftrightarrow \{i \in I \Rightarrow 1 - v_i = w_i\}$

$$\#I - \left(\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1 \right) \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1$$

\Leftrightarrow

$$\delta(C) - \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor - 1 \leq \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 1$$

\Leftrightarrow

$$\delta(C) \leq 2 \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor + 2$$

$\Leftrightarrow \{\frac{\delta(C)-1}{2} \text{ es entero o } 0.5\}$

$$\delta(C) \leq 2 \left(\frac{\delta(C)-1}{2} - 0.5 \right) + 2$$

\Leftrightarrow

$$\delta(C) \leq \delta(C) - 1 - 1 + 2$$

\Leftrightarrow

$$\delta(C) \leq \delta(C)$$

\Leftrightarrow

True

Cota de Hamming:

Sea C un código de longitud n

$$\#C \leq \frac{2^n}{\sum_{r=0}^{\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor} \binom{n}{r}}$$

Demostración:

Sea:

$$t = \left\lfloor \frac{\delta(C) - 1}{2} \right\rfloor$$

$$\#C \leq \frac{2^n}{\sum_{r=0}^t \binom{n}{r}}$$

\Leftrightarrow

$$\#C \sum_{r=0}^t \binom{n}{r} \leq 2^n$$

\Leftrightarrow

$$\sum_{u \in C} \sum_{r=0}^t \binom{n}{r} \leq 2^n$$

$\Leftrightarrow \{\#D_t(u) = \sum_{r=0}^t \binom{n}{r} \text{ ya que es las formas de elegir } 0, \dots, t \text{ bits de } u \text{ y cambiarlos}\}$

$$\sum_{u \in C} \#D_t(u) \leq 2^n$$

$\Leftrightarrow \{\text{Por teorema } C \text{ corrige } t \text{ errores} \Rightarrow \langle \forall u, v \in C : D_t(u) \cap D_t(v) = \emptyset \rangle \Rightarrow \text{el cardinal de la union es la suma de los cardinales}\}$

$$\# \bigcup_{u \in C} D_t(u) \leq 2^n$$

$\Leftarrow \{\#\{0,1\}^n = 2^n\}$

$$\bigcup_{u \in C} D_t(u) \subseteq \{0,1\}^n$$

\Leftarrow

$$\forall u \in C : D_t(u) \subseteq \{0,1\}^n$$

$\Leftrightarrow \{\text{Esto es claramente cierto}\}$

True

δ es el menor peso de Hamming en códigos lineales:

Sean:

$$v, w \in \{0,1\}^n$$

C un código lineal

$$\delta(C) = \min\{|u| : u \in C - \{0\}\}$$

Demostración por doble desigualdad:

Pruebo el \geq :

$$\begin{aligned} & \delta(C) \\ &= \min\{d(s, v) : s, v \in C : s \neq v\} \\ &= \{d(s, u) = |s + u|\} \\ & \quad \min\{|s + v| : s, v \in C : s \neq v\} \\ &\geq \{s + v \in C, s \neq 0 \vee v \neq 0 \Rightarrow s + v \neq 0\} \\ & \quad \min\{|u| : u \in C - \{0\}\} \end{aligned}$$

Pruebo el \leq :

$$\begin{aligned} \delta(C) &\leq \min\{|u| : u \in C - \{0\}\} \\ \Leftrightarrow \forall u \in C - \{0\} : \delta(C) &\leq |u| \end{aligned}$$

Trabajo sin el \forall :

$$\begin{aligned} \delta(C) &\leq |u| \\ \Leftrightarrow \delta(C) &\leq |u + 0| \\ \Leftrightarrow \{d(s, u) = |s + u|\} &\\ \delta(C) &\leq d(u + 0) \\ \Leftrightarrow \{\text{Por definición de } \delta, \delta(C) \text{ es la mínima distancia (y } 0 \in C\} & \\ \text{True} & \end{aligned}$$

Matriz de chequeo:

Sean:

$$n, k \in \mathbb{N}$$

C un código lineal

$$A, B \in \{0,1\}^{k \times n-k}$$

$[\text{Id}_k \ A], [B \ \text{Id}_k]$ generan a C

$[A^t \ \text{Id}_{n-k}], [\text{Id}_{n-k} \ B^t]$ son matrices de chequeo de C

Demostración $[A^t \ \text{Id}_{n-k}]$ es matriz de chequeo de C :

$$\begin{aligned} &[A^t \ \text{Id}_{n-k}] \text{ es matriz de chequeo de } C \\ \Leftrightarrow &\{\text{Sea: } T(\alpha) = [A^t \ \text{Id}_{n-k}]\alpha^t\} \\ &\text{Nu}(T) = C \\ \Leftrightarrow &\forall \alpha \in \{0,1\}^n : T(\alpha) = 0 \Leftrightarrow \alpha \in C \end{aligned}$$

Trabajo sin el \forall , demuestro ida y vuelta por separado:

Vuelta (\Rightarrow) suponiendo el antecedente:

Sea:

$$\alpha \in C$$

$\beta \in \{0,1\}^k$ tal que $\alpha = \beta[\text{Id}_k \ A]$ (existe porque $[\text{Id}_k \ A]$ genera a C)

$$\begin{aligned} &T(\alpha) \\ = &[A^t \ \text{Id}_{n-k}]\alpha^t \\ = &[A^t \ \text{Id}_{n-k}](\beta[\text{Id}_k \ A])^t \\ = &[A^t \ \text{Id}_{n-k}][\text{Id}_k \ A]^t \beta^t \\ = & \end{aligned}$$

$$\begin{aligned}
& [A^t \quad \text{Id}_{n-k}] \begin{bmatrix} \text{Id}_k^t \\ A^t \end{bmatrix} \beta^t \\
= & [A^t \quad \text{Id}_{n-k}] \begin{bmatrix} \text{Id}_k \\ A^t \end{bmatrix} \beta^t \\
= & (A^t \text{Id}_k + \text{Id}_{n-k} A^t) \beta^t \\
= & (A^t + A^t) \beta^t \\
= & 0 \beta^t \\
= & 0
\end{aligned}$$

Ida (\Rightarrow):

Sea:

$$[\alpha' \quad \alpha''] = \alpha$$

Con $\alpha' \in \{0,1\}^k$, $\alpha'' \in \{0,1\}^{n-k}$

$$\begin{aligned}
& T(\alpha) = 0 \\
\Leftrightarrow & [A^t \quad \text{Id}_{n-k}] \alpha^t = 0 \\
\Leftrightarrow & [A^t \quad \text{Id}_{n-k}] [\alpha' \quad \alpha'']^t = 0 \\
\Leftrightarrow & [A^t \quad \text{Id}_{n-k}] \begin{bmatrix} \alpha'^t \\ \alpha''^t \end{bmatrix} = 0 \\
\Leftrightarrow & A^t \alpha'^t + \text{Id}_{n-k} \alpha''^t = 0 \\
\Leftrightarrow & A^t \alpha'^t + \alpha''^t = 0 \\
\Leftrightarrow & A^t \alpha'^t = \alpha''^t \\
\Leftrightarrow & \alpha' A = \alpha''^t
\end{aligned}$$

Ahora uso esto para probar suponiendo el antecedente:

$$\alpha \in C$$

$$\begin{aligned}
& \Leftrightarrow \exists \beta \in \{0,1\}^k : \alpha = \beta [\text{Id}_k \quad A]
\end{aligned}$$

Si existe:

$$\beta = \alpha'$$

$$\begin{aligned}
& \alpha = \alpha' [\text{Id}_k \quad A] \\
\Leftrightarrow & \{\text{Por lo probado anteriormente: } \alpha = [\alpha' \quad \alpha' A]\} \\
& [\alpha' \quad \alpha' A] = \alpha' [\text{Id}_k \quad A] \\
\Leftrightarrow & [\alpha' \quad \alpha' A] = [\alpha' \text{Id}_k \quad \alpha' A] \\
\Leftrightarrow & [\alpha' \quad \alpha' A] = [\alpha' \quad \alpha' A] \\
\Leftrightarrow & \text{True}
\end{aligned}$$

Demostración $[\text{Id}_{n-k} \quad B^t]$ es matriz de chequeo de C :

$[\text{Id}_{n-k} \quad B^t]$ es matriz de chequeo de C

$$\Leftrightarrow \{ \text{Sea: } T(\alpha) = [\text{Id}_{n-k} \quad B^t] \alpha^t \}$$

$$\text{Nu}(T) = C$$

$$\Leftrightarrow$$

$$\forall \alpha \in \{0,1\}^n : T(\alpha) = 0 \Leftrightarrow \alpha \in C$$

Trabajo sin el \forall , demuestro ida y vuelta por separado:

Vuelta (\Leftarrow) suponiendo el antecedente:

Sea:

$$\alpha \in C$$

$\beta \in \{0,1\}^k$ tal que $\alpha = \beta[B \quad \text{Id}_k]$ (existe porque $[B \quad \text{Id}_k]$ genera a C)

$$\begin{aligned} & T(\alpha) \\ = & [\text{Id}_{n-k} \quad B^t] \alpha^t \\ = & [\text{Id}_{n-k} \quad B^t] (\beta[B \quad \text{Id}_k])^t \\ = & [\text{Id}_{n-k} \quad B^t] [B \quad \text{Id}_k]^t \beta^t \\ = & [\text{Id}_{n-k} \quad B^t] \begin{bmatrix} B^t \\ \text{Id}_k^t \end{bmatrix} \beta^t \\ = & [\text{Id}_{n-k} \quad B^t] \begin{bmatrix} B^t \\ \text{Id}_k \end{bmatrix} \beta^t \\ = & (\text{Id}_{n-k} B^t + B^t \text{Id}_k) \beta^t \\ = & (B^t + B^t) \beta^t \\ = & 0 \beta^t \\ = & 0 \end{aligned}$$

Ida (\Rightarrow):

Sea:

$$[\alpha' \quad \alpha''] = \alpha$$

Con $\alpha' \in \{0,1\}^{n-k}$, $\alpha'' \in \{0,1\}^k$

$$\begin{aligned} & T(\alpha) = 0 \\ \Leftrightarrow & [\text{Id}_{n-k} \quad B^t] \alpha^t = 0 \\ \Leftrightarrow & [\text{Id}_{n-k} \quad B^t] [\alpha' \quad \alpha'']^t = 0 \\ \Leftrightarrow & [\text{Id}_{n-k} \quad B^t] \begin{bmatrix} \alpha'^t \\ \alpha''^t \end{bmatrix} = 0 \\ \Leftrightarrow & \text{Id}_{n-k} \alpha'^t + B^t \alpha''^t = 0 \\ \Leftrightarrow & \alpha'^t + B^t \alpha''^t = 0 \\ \Leftrightarrow & B^t \alpha''^t = \alpha'^t \\ \Leftrightarrow & \alpha'' B = \alpha' \end{aligned}$$

Ahora uso esto para probar suponiendo el antecedente:

$$\alpha \in C$$

\Leftrightarrow

$$\exists \beta \in \{0,1\}^k : \alpha = \beta[B \quad \text{Id}_k]$$

Si existe:

$$\beta = \alpha''$$

$$\alpha = \alpha''[B \quad \text{Id}_k]$$

\Leftrightarrow {Por lo probado anteriormente: $\alpha = [\alpha''B \quad \alpha'']$ }

$$[\alpha''B \quad \alpha''] = \alpha''[B \quad \text{Id}_k]$$

\Leftrightarrow

$$[\alpha''B \quad \alpha''] = [\alpha''B \quad \alpha''\text{Id}_k]$$

\Leftrightarrow

$$[\alpha''B \quad \alpha''] = [\alpha''B \quad \alpha'']$$

\Leftrightarrow

True

Delta con matriz de chequeo:

Sean:

C un código lineal de longitud n

H una matriz de chequeo de C

$$\delta(C) = \min\{\#D : D \subseteq \text{columnas}(H) \wedge D \text{ es LD}\}$$

Demostración:

$$\delta(C) = \min\{\#D : D \subseteq \text{columnas}(H) \wedge D \text{ es LD}\}$$

\Leftrightarrow {Teorema: $\delta(C) = \min\{|u| : u \in C - \{0\}\}$ }

$$\min\{|u| : u \in C - \{0\}\} = \min\{\#D : D \subseteq \text{columnas}(H) \wedge D \text{ es LD}\}$$

Pruebo esto por doble desigualdad:

(\leq): Para $u \in C - \{0\}$ doy un D con $|u| \leq D$

Sea:

$$u \in C - \{0\}$$

$$r = |u|$$

$j_1, j_2, \dots, j_r \in \mathbb{N}_{\leq n}$ tal que $u_{j_1}, u_{j_2}, \dots, u_{j_r} = 1$ (j_1, j_2, \dots, j_r son los 1s de u , y son todos números distintos)

Sea:

$$D = \{\text{columna}_i(H) : i \in \{j_1, j_2, \dots, j_r\}\}$$

Pruebo $D \subseteq \text{columnas}(H) \wedge D \text{ es LD} \wedge \#D \leq r$

Es claro que $D \subseteq \text{columnas}(H)$ y $\#D = r$, pruebo D es LD:

Por ser H matriz de chequeo de C :

$$Hu^t = 0$$

\Leftrightarrow

$$\sum_{i=1}^n \text{columna}_i(H)u_i = 0$$

\Leftrightarrow {Sumo solo los índices con $u_i = 1$ }

$$\begin{aligned}
& \sum_{i \in \{j_1, j_2, \dots, j_r\}} \text{columna}_i(H) u_i = 0 \\
\Leftrightarrow & \{\text{Estos } u_i \text{ son } 1\} \\
& \sum_{i \in \{j_1, j_2, \dots, j_r\}} \text{columna}_i(H) = 0 \\
\Leftrightarrow & \{\text{Definición LD}\} \\
& \{\text{columna}_i(H) : i \in \{j_1, j_2, \dots, j_r\}\} \text{ es LD} \\
\Leftrightarrow & \{\text{Definición } D\} \\
& D \text{ es LD}
\end{aligned}$$

(\geq): Para un D que minimiza $|u|$ con $|u| \geq D$

Sea:

$$D = \min_{\substack{D' \subseteq \text{columnas}(H) : D' \text{ es LD}}} \#D'$$

(O sea D minimiza $\#D$)

$$r = \#D$$

j_1, j_2, \dots, j_r los índices de columna que forman D

Es decir, $D = \{\text{columna}_i(H) : i \in \{j_1, j_2, \dots, j_r\}\}$

$u \in \{0,1\}^n$ definido de la siguiente manera:

$$u_i = \begin{cases} i \in \{j_1, j_2, \dots, j_r\} & \rightarrow 1 \\ \text{si no} & \rightarrow 0 \end{cases}$$

Pruebo que $u \in C$

$$\begin{aligned}
& u \in C \\
\Leftrightarrow & Hu^t = 0 \\
\Leftrightarrow & \sum_{i=1}^n u_i \text{columna}_i(H) = 0 \\
\Leftrightarrow & \sum_{i=1}^r \text{columna}_{j_r}(H) = 0 \\
\Leftrightarrow & \{j_1, j_2, \dots, j_r\} \text{ son los índices de columna que forman } D, \text{ y como } D \text{ minimiza } \#D, \text{ ningún subconjunto suyo es LD} \\
0 & = 0 \\
\Leftrightarrow & \text{True}
\end{aligned}$$

Pruebo que $|u| \geq \#D$

$$\begin{aligned}
& |u| \\
= & \{\text{Por como está definido } u\} \\
& \#\{j_1, j_2, \dots, j_r\} \\
= & \{\text{Cada uno de estos índices produce un elemento de } D\} \\
& \#D
\end{aligned}$$

Tamaño de matriz de chequeo:

Sea:

C un código lineal de longitud n

H una matriz de chequeo de C

$H \in \{0,1\}^{r \times n}$

Las filas de H son LI $\Leftrightarrow r = n - \dim(C)$

Demostración:

Sea:

$T : \{0,1\}^n \rightarrow \{0,1\}^r$

$T(\alpha) = H\alpha^t$

Las filas de H son LI

\Leftrightarrow

$\dim(\text{spacioFila}(H)) = r$

$\Leftrightarrow \{\text{Teorema de álgebra}\}$

$\dim(\text{spacioColumna}(H)) = r$

$\Leftrightarrow \{\text{Teorema de álgebra}\}$

$\dim(\text{Im}(T)) = r$

$\Leftrightarrow \{\text{Teorema: } \dim(\{0,1\}^n) = \dim(\text{Im}(T)) \geq \dim(\text{Ker}(T))\}$

$\dim(\{0,1\}^n) - \dim(\text{Ker}(T)) = r$

$\Leftrightarrow \{H \text{ es matriz de chequeo de } C\}$

$n - \dim(C) = r$

\Leftrightarrow

$r = n - \dim(C)$

Cota de Singleton:

Sea:

C un código lineal de longitud n

$$\delta(C) \leq n - \dim(C) + 1$$

Demostración:

Sea:

$H \in \{0,1\}^{r \times n}$

H una matriz de chequeo con filas LI

En primer lugar:

las filas de H son LI

\Leftrightarrow

$\dim(\text{spacioColumna}(H)) = r$

\Leftrightarrow

$\forall D \subseteq \text{columnas}(H) : D \text{ es LI} : \#D = r$

Así que sea $D \subseteq \text{columnas}(H)$ tal que D es LI

Hago la demostración por contradicción, o sea, supongo $\delta(C) > n - \dim(C) + 1$:

$$\delta(C) > n - \dim(C) + 1$$

$\Leftrightarrow \{\text{Por teorema: Las filas de } H \text{ son LI} \Rightarrow r = n - \dim(C)\}$

$$\delta(C) > r + 1$$

\Leftrightarrow

$\delta(C) - 1 > r$
 $\Leftrightarrow \{\text{Primera demostración}\}$
 $\delta(C) - 1 > \#D$
 $\Rightarrow \{\text{Teorema: } \delta(C) = \min\{\#D : D \subseteq \text{columnas}(H) \wedge D \text{ es LD}\} \Rightarrow \#D \geq \delta(C)\}$
 $\delta(C) - 1 > \delta(C)$
 \Rightarrow
False

I): Encontrar la distancia de Hamming entre x e y :

a) $x = 1110010, y = 0101010$ b) $x = 11100011, y = 10110001$

d) $x = \begin{array}{ccccccc} 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ y = \begin{array}{ccccccc} 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array} \end{array}$

$$d(x, y) = 3$$

b) $x = \begin{array}{ccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ y = \begin{array}{ccccccc} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array} \end{array}$

$$d(x, y) = 3$$

1)

miércoles, 4 de mayo de 2022 17:27

I): En cada uno de los códigos siguientes, encontrar el número de errores que pueden ser detectados/corregidos.

a) $C = \{0000000, 1111110, 1010100, 0101010\}$

b) $C = \{0000000, 1111111, 1111000, 0000111\}$

c) $C = \{000000000, 111111111, 111100000, 000011111, 101010101, 010101010\}$

Por teorema:

Sea C un código del longitud n

C detecta hasta $\delta(C) - 1$ errores

C corrige hasta $\left\lfloor \frac{\delta(C) - 1}{2} \right\rfloor$ errores

$$\text{2)} \quad C = \{0000000, 1111110, 1010100, 0101010\}$$

$$\begin{matrix} a & b & c & d \end{matrix}$$

$$\delta(C) = \min \{d(x, y) : x, y \in C : x \neq y\}$$

$$= \min \{d(a, b), d(a, c), d(a, d), d(b, c), d(b, d), d(c, d)\}$$

$$= \min \{6, 3, 3, 3, 3, 6\}$$

$$= 3$$

\Rightarrow

C detecta hasta

$$= \delta(C) - 1$$

$$= 3 - 1$$

$$= 2$$

C corrige hasta

$$\left\lfloor \frac{\delta(C) - 1}{2} \right\rfloor$$

$$\approx \left\lfloor \frac{3-1}{2} \right\rfloor$$

$$\approx 1$$

b) $C = \{0000000, 1111111, 1111000, 0000111\}$

$$\delta(C) = \min\{6, 3, 3, 3, 6\} = 3$$

\Rightarrow

C detecta hasta 2

C corrige hasta 1

c) $C = \{000000000, 111111111, 111100000, 000011111, 101010101, 010101010\}$

$$\delta(C) = \min\{9, 9, 5, 5, 4, 5, 4, 4, 5, 9, 5, 4, 4, 5, 9\}$$

$$\approx 4$$

\Rightarrow C detecta hasta

$$= 4 - 1 = 3$$

C corrige hasta

$$\approx \left\lfloor \frac{4-1}{2} \right\rfloor = 2$$

2)

viernes, 6 de mayo de 2022 14:26

II): Se tiene un código binario de longitud 8 tal que la distancia mínima entre palabras es 5. Dar una cota superior para el número de palabras del código. Dar una cota superior si además se requiere que el código sea lineal.

Sea:

 C un código de longitud 8

$$\delta(C) = 5$$

$$\#C \leq 6$$

$$C \text{ es lineal} \Rightarrow \#C \leq 4$$

Demostraciones:

$$\begin{aligned} & \#C \\ \leq & \frac{2^n}{\sum_{r=0}^{\lfloor \frac{\delta(C)-1}{2} \rfloor} \binom{n}{r}} \\ = & \frac{2^8}{\sum_{r=0}^{\lfloor \frac{5-1}{2} \rfloor} \binom{8}{r}} \\ = & \frac{2^8}{\sum_{r=0}^2 \binom{8}{r}} \\ = & \frac{2^8}{\binom{8}{0} \binom{8}{1} \binom{8}{2}} \\ = & \frac{2^8}{1 + 2 + 28} \\ \approx & 6.9189 \end{aligned}$$

Y como $\#C$ es entero, $\#C \leq 6$ C es lineal:

$$\begin{aligned}\#C & \\ \leq & \\ 2 & \left\lceil \log_2 \left(\frac{2^n}{\sum_{r=0}^{\lfloor \delta(C)-1 \rfloor} \frac{n^2}{r!}} \right) \right\rceil \\ = & \\ 2^{\lfloor \log_2(6.9189) \rfloor} & \\ = & \\ 4 &\end{aligned}$$

3)

viernes, 6 de mayo de 2022 14:42

III): Se tiene un código binario de longitud 16 tal que la distancia mínima entre palabras es 7. Dar una cota superior para el número de palabras del código. Dar una cota superior si además se requiere que el código sea lineal.

Sea:

 C un código de longitud 16

$$\delta(C) = 7$$

$$\#C \leq 94$$

$$C \text{ es lineal} \Rightarrow \#C \leq 64$$

$$\begin{aligned} & \#C \\ \leq & \frac{2^{16}}{\sum_{r=0}^{\lfloor \frac{\delta(C)-1}{2} \rfloor} \binom{16}{r}} \\ = & \frac{2^{16}}{\sum_{r=0}^{\lfloor \frac{7-1}{2} \rfloor} \binom{16}{r}} \\ = & \frac{2^{16}}{\sum_{r=0}^3 \binom{16}{r}} \\ = & \frac{2^{16}}{\binom{16}{0} + \binom{16}{1} + \binom{16}{2} + \binom{16}{3}} \\ = & \frac{2^{16}}{1 + 16 + \frac{16 * 15}{2} + \frac{16 * 15 * 14}{3 * 2}} \\ \approx & 94.0258 \end{aligned}$$

Sea C lineal

$$\begin{aligned} & \#C \\ \leq & 2^{\left\lceil \log_2 \left(\frac{2^n}{\sum_{r=0}^{\lfloor \frac{\delta(C)-1}{2} \rfloor} r!} \right) \right\rceil} \\ = & \end{aligned}$$

$$2^{\lfloor \log_2(94.0258) \rfloor}$$

=

$$64$$

IV): Probar que todo código binario lineal con parámetros $(n, k, \delta) = (23, 11, 7)$ es perfecto.

Sea:

C un código lineal de parámetros $(23, 12, 7)$ (los profes dijeron que el PDF tenía un error)

c es perfecto

Demostración:

c es perfecto

\Leftrightarrow

$$\#C = \frac{2^n}{\sum_{r=0}^{\lfloor \frac{\delta(C)-1}{2} \rfloor} \binom{n}{r}}$$

$\Leftrightarrow \{\#C = 2^k, n = 23, k = 12, \delta(C) = 7\}$

$$2^{12} = \frac{2^{23}}{\sum_{r=0}^{\lfloor \frac{7-1}{2} \rfloor} \binom{23}{r}}$$

\Leftrightarrow

$$2^{12} = \frac{2^{23}}{\sum_{r=0}^3 \binom{23}{r}}$$

\Leftrightarrow

$$2^{12} = \frac{2^{23}}{\binom{23}{0} \binom{23}{1} \binom{23}{2} \binom{23}{3}}$$

\Leftrightarrow

$$2^{12} = \frac{2^{23}}{1 + 23 + \frac{23 * 22}{2} + \frac{23 * 22 * 21}{3 * 2}}$$

\Leftrightarrow

$$4096 = 4096$$

\Leftrightarrow

True

5)

miércoles, 4 de mayo de 2022 15:07

V): Sea

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

y sea C el código generado por G .

- a) ¿Cuál es la longitud de C ? Cuál es su dimensión?
 b) Supongamos que desea mandar el mensaje 10101 11010 00111. ¿Cuáles son las palabras del código que debería usar para mandar el mensaje?
 b) Dar una matriz de chequeo de C .
 c) Calcular $\delta(C)$.
 d) Supongamos que se reciben las palabras 100111001 y 110000001. ¿Cuáles son las palabras más probables que se hayan mandado? ¿A qué mensaje corresponden?
 e) ¿Qué puede concluir si recibe la palabra 001111010? ¿Y si recibe la palabra 011100011?

$$C = \text{espacioFila}(G)$$

a)

$$\dim(G)$$

=

$$\text{rangoFila}(G)$$

= {Las filas son LI, porque hay una Id_5 en las primeras 5 columnas}

$$5$$

$$\#C$$

= { C es lineal}

$$2^{\dim(G)}$$

=

$$2^5$$

=

$$32$$

b)

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

b)

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

$$= [1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1]$$

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

$$= [1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]$$

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

$$= [0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0]$$

b')

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

$$G = [\text{Id}_5 \quad A]$$

Por teorema:

$[A^t \quad \text{Id}_4]$ es matriz de chequeo de C

$$H = [A^t \quad \text{Id}_4] = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

c)

Por teorema:

$$\delta(C) = \min\{\#D : D \subseteq \text{columnas}(H) \wedge D \text{ es LD}\}$$

En este caso hay varios conjuntos de 3, por ejemplo $\{\text{columna}_1(H), \text{columna}_6(H), \text{columna}_7(H)\}$, pero no hay ninguna de 2 porque no hay dos filas iguales

d)

$$\alpha = 100111001$$

$$H\alpha = \begin{bmatrix} 1 + 1 + 0 + 1 + 0 \\ 1 + 1 + 1 + 0 + 0 \\ 0 + 1 + 0 + 0 + 0 \\ 0 + 0 + 1 + 0 + 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$= \text{columna}_4(H)$

\Rightarrow

El mensaje mas probable es 100011001

\Rightarrow

El mensaje original mas probable es 10001

$$\beta = 110000001$$

$$H\beta = \begin{bmatrix} 1 + 1 + 0 \\ 1 + 0 + 0 \\ 0 + 1 + 0 \\ 0 + 0 + 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$= \text{columna}_3(H)$

\Rightarrow

El mensaje mas probable es 111000001

⇒

La palabra original mas probable es 11100

e)

$$\alpha = 001111010$$

$$H\alpha = \begin{bmatrix} 0 + 1 + 0 + 1 + 0 \\ 1 + 1 + 1 + 0 + 0 \\ 1 + 1 + 0 + 0 + 1 \\ 1 + 0 + 1 + 0 + 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$\notin \text{columnas}(H)$

⇒

Hubo un error, pero no hay una palabra original mas probable

$$\beta = 011100011$$

$$H\beta = \begin{bmatrix} 1 + 0 + 1 + 0 + 0 \\ 0 + 1 + 1 + 0 + 0 \\ 1 + 1 + 1 + 1 + 0 \\ 0 + 1 + 0 + 0 + 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

⇒

El mensaje se trasmitió correctamente

⇒

La palabra original es 01110

VI): Sea H la matriz de chequeo:

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

y sea C el código asociado a ella.

- a) Describir C explicitamente (es decir, dar las palabras que constituyen el código).
b) Calcular $\delta(C)$.

c) Suponga que Ud. recibe la palabra 00111000. Asumiendo que se produjo a lo sumo un error de transmisión, ¿que palabra le fue enviada?

d) Ud. recibe la palabra 11100111. ¿Que puede concluir?

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

$$H = [\text{Id}_4 \quad B]$$

H es matriz de chequeo de C

Por teorema:

$[B^t \quad \text{Id}_4]$ es matriz generadora de C

$$G = [B^t \quad \text{Id}_4] = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$C = \text{espacioFila}(G)$

a)

$$\begin{aligned} C = \{ & 00000000, 11011000, 01010100, 10001100, 11100010, 00111010, 10110110, 01101110, \\ & 01100001, 10111001, 00110101, 11101101, 10000011, 01011011, 11010111, 00001111 \} \end{aligned}$$

b)

Por teorema:

$$\delta(C) = \min\{\#D : D \subseteq \text{columnas}(H)\}$$

En este caso es 3, porque hay varios de 3, por ejemplo las columnas 4, 5 y 6, pero no hay de 2, porque no hay 2 columnas iguales

c)

$$\alpha = 00111000$$

$$\begin{aligned} H\alpha &= \begin{bmatrix} 0+0+1 \\ 0+0+1 \\ 1+0+0 \\ 0+1+1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \\ &= \text{columna}_7(H) \end{aligned}$$

\Rightarrow

Fue enviada 00111010, proveniente de 1010

d)

$$\alpha = 11100111$$

$$H\alpha = \begin{bmatrix} 1 + 0 + 0 + 0 + 1 + 0 \\ 0 + 1 + 0 + 1 + 1 + 1 \\ 0 + 0 + 1 + 0 + 1 + 1 \\ 0 + 0 + 0 + 1 + 0 + 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$\notin \text{columnas}(H)$

\Rightarrow

Hubo mas de 1

7)

miércoles, 11 de mayo de 2022 11:45

IX): Dar un ejemplo de un código lineal C con matriz generadora G tal que $\delta(C)$ NO sea igual a la menor cantidad de unos que aparece en alguna fila de G

$$C = \text{especioFila}(G)$$

$$\delta(C) \neq \min\{|v| : v \text{ es una fila de } G\}$$

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

$$\begin{aligned} C &= \text{espacioFila}(G) \\ &= \{000, 111, 011, 100\} \end{aligned}$$

$$\delta(c) = 1$$

$$\min\{|v| : v \text{ es una fila de } G\} = 2$$

VIII): Sea H la matriz de chequeo:

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

y sea C el código asociado a ella.

- a) Escribir 5 palabras que estén en C . ¿Cuántas palabras tiene en total C ?
- b) Calcular $\delta(C)$.
- c) Suponga que Ud. recibe la palabra 11100000000011. Asumiendo que se produjo a lo sumo un error de transmisión, ¿qué palabra le fue enviada?

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

$$H = [\text{Id}_5 \quad A]$$

H es matriz de chequeo de C

a)

$$\text{Sea } G = [A^t \quad \text{Id}_9]$$

Por teorema, G es matriz generadora de C

Algunas palabras en el código son:

$$0G = 0$$

$$e_1G = 11000100000000$$

$$e_2G = 10101010000000$$

$$e_3G = 01101001000000$$

$$e_4G = 11100000100000$$

Por teorema:

C tiene en total 2^9 elementos

c)

Por teorema:

$$\delta(C) = \min\{\#D : D \subseteq \text{columnas}(H)\}$$

En este caso es 3, ya que hay varios de 3, por ejemplo, las columnas 1, 2 y 6, pero no hay ninguno de 2, por que no hay 2 columnas iguales

d)

$$\alpha = 11100000000011$$

$$H\alpha = \begin{bmatrix} 1 + 0 + 0 + 0 + 1 \\ 0 + 1 + 0 + 0 + 0 \\ 0 + 0 + 1 + 1 + 1 \\ 0 + 0 + 0 + 1 + 1 \\ 0 + 0 + 0 + 0 + 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$= \text{columna}_8(H)$

Lo mas probable es que se haya mandado 11100001000011

IX):

Sea C el código con matriz de chequeo:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & a & b \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & c & d \end{bmatrix}$$

Escribir dos palabras no nulas que estén en C , decir cuantas palabras tiene en total C , calcular $\delta(C)$, justificando y, si se recibe la palabra 100000000000111, y se asume que se produjo a lo sumo un error de transmisión, determinar la palabra enviada si esto es posible o indicar porqué no si no se puede. Las respuestas pueden depender de cuales valores toman $a, b, c, d \in \{0, 1\}$, si es así, ud. debe indicarlo y dar todas las respuestas posibles.

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & a & b \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & c & d \end{bmatrix}$$

$$H = [\text{Id}_6 \quad A]$$

H es matriz de chequeo de C

Sea:

$$G = [A^t \quad \text{Id}_{10}]$$

Por teorema:

 G genera a C Algunas palabras en C son:

$$e_1 G = 1111001000000000$$

$$e_2 G = 1111100100000000$$

C tiene 2^{10} palabras en total

Por teorema:

$$\delta(C) = \min\{\#D : D \subseteq \text{columnas}(H)\}$$

\Rightarrow

$$\delta(C) = \begin{cases} a = c = 1 & \rightarrow 2 \\ \text{si no} & \rightarrow 3 \end{cases}$$

Si $a = c = 1, D = \{\text{columna}_{12}(H), \text{columna}_{15}(H)\}$

Si no, $D = \{\text{columna}_6(H), \text{columna}_7(H), \text{columna}_8(H)\}$

$$\alpha = 1000000000000111$$

$$H\alpha = \begin{bmatrix} 1 + 1 + 0 + 1 \\ 0 + 1 + 0 + 0 \\ 0 + 0 + 1 + 1 \\ 0 + 1 + 1 + 1 \\ 0 + 0 + a + b \\ 0 + 0 + c + d \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ a + b \\ c + d \end{bmatrix}$$

Si $a + b = 0 \wedge c + d = 0$:

$$H\alpha = \text{columna}_{14}(H)$$

\Rightarrow

Se mando 1000000000000011

Si $a + b = 0 \wedge c + d = 1$:

$$H\alpha = \text{columna}_{11}(H)$$

\Rightarrow

Se mando 1000000000100111

Si $a + b = 1 \wedge c + d = 1$:

$$H\alpha = \text{columna}_{10}(H)$$

\Rightarrow

Se mando 1000000001000111

Si $a + b = 1 \wedge c + d = 0$:

$$H\alpha = \text{columna}_8(H)$$

⇒

Se mando 1000000100000111

10)

domingo, 15 de mayo de 2022 8:34

X): Una empresa necesita codificar un millón de palabras. Desea que el código sea capaz de corregir un error.

- Supongamos que Ud. desea diseñar un código lineal por medio de una matriz de chequeo que satisfaga esto. ¿Cuál es el menor tamaño que debe tener la matriz?
- Escriba una matriz que satisfaga las condiciones, del tamaño dado en a).
- Escriba dos palabras de peso menor o igual a 6 que estén en su código y una palabra de peso mayor o igual a 15 que esté en su código.
- Tome una de sus palabras de b), y cambie los dos primeros dígitos. (si es un 1, escriba 0 y viceversa) Suponga que esa es la palabra que se recibe. Prediga que deducirá la persona que la recibe, de acuerdo con el código diseñado por Ud. Explique bien por qué.

a)

Los números entre 0 y 999999 se pueden codificar con:

$$\begin{aligned} k &= \lceil \log_2(1000000) \rceil \\ &= 20 \end{aligned}$$

Por lo cuál, hace falta que $\dim(C) = 20$

Para corregir un error hace falta que $\delta(C) \geq 3$

Sea:

n la longitud del código

Por teorema:

$$\begin{aligned} \delta(C) &\leq n - \dim(C) + 1 \\ \Leftrightarrow 3 &\leq n - 20 + 1 \\ \Leftrightarrow 3 + 20 - 1 &\leq n \\ \Leftrightarrow 22 &\leq n \end{aligned}$$

También, por el límite de la relación entre alto y alto de la matriz de chequeo para que las filas se mantengan LI:

$$\begin{aligned} n &\leq 2^{n-k} - 1 \\ \Leftrightarrow n &\leq 2^{n-20} - 1 \end{aligned}$$

El mínimo n que cumple esto es 25, ya que:

$$\begin{aligned} 25 &< 31 = 32 - 1 = 2^5 - 1 = 2^{25-20} - 1 \\ 24 &\not\leq 15 = 16 - 1 = 2^4 - 1 = 2^{24-20} - 1 \end{aligned}$$

Así que el mínimo tamaño de matriz de chequeo es 5×25 :

$$H \in \{0,1\}^{5 \times 25}$$

b)

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

$$H = [\text{Id}_5 \quad A]$$

c)

$$G = [A^t \quad \text{Id}_{20}]$$

Palabras de paso menor o igual a 6:

$$\alpha = 10000000000000000000G = 11000100000000000000000000000000$$

$$\beta = 01000000000000000000G = 10100010000000000000000000000000$$

palabra con peso mayor o igual a 15:

$$\gamma = 111111111111111111G = 000011111111111111111111$$

d)

$$\alpha' = 00000100000000000000000000000000$$

$$H\alpha'^t = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$= \text{columna}_6(H)$

⇒ El que la reciba creera que se mandó la palabra 00000000000000000000000000000000

XI): Una consultora realizará 65 preguntas a una población. Cada pregunta tendrá como respuestas posibles "Siempre", "Frecuentemente", "De vez en cuando", "Rara Vez", y "Nunca". La compañía quiere codificar esta información. (por lo tanto, los datos a codificar son cosas del tipo "pregunta 32, respuesta Frecuentemente"). La encuestadora desea que el código sea capaz de corregir un error por dato y que codifique todos los datos posibles.

- Diseñe un código lineal que satisfaga esto, dando una matriz de chequeo apropiada del menor tamaño posible.
- Escriba dos palabras que estén en su código.
- Tome una de sus palabras de b), y cambie los dos primeros dígitos. Suponga que esa es la palabra que se recibe. Prediga qué deducirá la persona que la recibe, de acuerdo con el código diseñado por Ud. Explique bien porqué.

a)

Hacen falta $65 * 5 = 325$ palabras

Por lo tanto:

$$k = \dim(C) = \lceil \log_2(325) \rceil = 9$$

C corrige un error $\Rightarrow \delta(C) \geq 3$

Sea n la longitud del código

Por teorema:

$$n \leq 2^{n-k} - 1$$

\Leftrightarrow

$$n \leq 2^{n-9} - 1$$

El menor n que satisface esto es 13, ya que:

$$13 < 15 = 16 - 1 = 2^4 - 1 = 2^{13-9} - 1$$

$$12 \not\leq 7 = 8 - 1 = 2^3 - 1 = 2^{12-9} - 1$$

Por ende:

Matriz generadora:

$$G \in \{0,1\}^{9 \times 13}$$

Matriz de chequeo

$$H \in \{0,1\}^{4 \times 13}$$

Sea:

$$H = \begin{bmatrix} \text{Id}_4 & \begin{matrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{matrix} \end{bmatrix}$$

$$T(\alpha) = H\alpha$$

$$C = \text{Nu}(T)$$

b)

$$\alpha = 1100100000000$$

$$\beta = 1010010000000$$

Están en el código:

$$H\alpha = \begin{bmatrix} 1+0+1 \\ 0+1+1 \\ 0+0+0 \\ 0+0+0 \\ 1+0+1 \end{bmatrix} = 0$$

$$H\beta = \begin{bmatrix} 0+0+0 \\ 0+0+0 \\ 0+1+1 \\ 0+0+0 \end{bmatrix} = 0$$

c)

$$\alpha' = 0000100000000$$

$$H\alpha' = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$= \text{columna}_5(H)$

\Rightarrow

La palabra en C mas cercana a α' es 0000000000000, por ende, el receptor va a creer que se mando 0000000000000

12)

martes, 17 de mayo de 2022 9:48

XII): Si tenemos un código de Hamming de 5 filas con la columna i -ésima igual a la representación binaria de i , contando columnas de izquierda a derecha y representando números con el bit menos significativo arriba, y llega la palabra

$$w = 0000101001000010000001000000001$$

calcular cual es la palabra mas probable enviada.

Uso $(x)_2^n$ para el vector de x en base 2, con n elementos

$$n = 31$$

$$k = 26$$

$$H = [(1)_2^5 \quad (2)_2^5 \quad \dots \quad (31)_2^5]$$

$$w = 0000101001000010000001000000001$$

$$\begin{aligned} Hw &= (5)_2^5 + (7)_2^5 + (10)_2^5 + (15)_2^5 + (22)_2^5 + (31)_2^5 \\ &= \begin{bmatrix} 1+1+0+1+0+1 \\ 0+1+1+1+1+1 \\ 1+1+0+1+1+1 \\ 0+0+1+1+0+1 \\ 0+0+0+0+1+1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = (14)_2^5 \\ &= \text{columna}_{14}(H) \end{aligned}$$

\Rightarrow

La palabra mas cercana es 0000101001000110000001000000001

13)

viernes, 6 de mayo de 2022 17:39

XIII): Sea A el conjunto de códigos de longitud 12 con 512 elementos, B el conjunto de códigos de longitud 12 con 3584 elementos y L el conjunto de códigos de longitud 12 que son lineales.

Probar que $|A| = |B|$ pero $|A \cap L| > |B \cap L|$.

$$A = \{A' \subseteq \{0,1\}^{12} : |A'| = 512\}$$

$$B = \{B' \subseteq \{0,1\}^{12} : |B'| = 3584\}$$

$$L = \{L' \subseteq \{0,1\}^{12} : L' \text{ es lineal}\}$$

$$\#A = \#B$$

$$\#(A \cap L) > \#(B \cap L)$$

Demostración:

$\#A = \#B$:

$$\begin{aligned} \#A &= \binom{2^{12}}{512} \\ &= \binom{4096}{4096 - 512} \\ &= \binom{2^{12}}{3584} \\ &= \#B \end{aligned}$$

$\#(A \cap L) > \#(B \cap L)$:

Primero pruebo $\#(B \cap L) = 0$:

$$\#(B \cap L) = 0$$

\Leftrightarrow

$$B \cap L = \emptyset$$

\Leftrightarrow

$$\nexists C \subseteq \{0,1\}^{12} : |C| = 3584 \wedge C \text{ es lineal}$$

$\Leftrightarrow \{\text{No existe } C \text{ porque } C \text{ es lineal} \Rightarrow |C| \text{ es potencia de 2}\}$

True

Ahora hago la demostración:

$$\begin{aligned} & \#(A \cap L) > \#(B \cap L) \\ \Leftrightarrow & \#(A \cap L) > 0 \\ \Leftrightarrow & A \cap L \neq \emptyset \\ \Leftrightarrow & \exists C \subseteq \{0,1\}^{12} : \#C = 512 \wedge C \text{ es lineal} \end{aligned}$$

Si existe:

Sea:

$$X = [\text{Id}_9 \quad 0] \in \{0,1\}^{9 \times 12}$$

$$C = \text{espcioFila}(X)$$

XIV): Probar que si C es un código binario que corrige un error pero no corrige 2, entonces C es perfecto si y solo si existe r tal que C tiene longitud $2^r - 1$ y 2^{2^r-r-1} elementos.

Sea:

C un código de longitud n

C corrige 1 error

C no corrige 2 errores

$$C \text{ es perfecto} \Leftrightarrow \exists r \in \mathbb{N} : n = 2^r - 1 \wedge \#C = 2^{2^r-r-1}$$

En primero lugar hago una demostración auxiliar:

C es perfecto

$\Leftrightarrow \{\text{Definición código perfecto}\}$

$$\#C = \frac{2^n}{\sum_{i=0}^{\left\lfloor \frac{\delta(C)-1}{2} \right\rfloor} \binom{n}{i}}$$

$\Leftrightarrow \{C \text{ corrige hasta } 1 \text{ error} \Rightarrow \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor = 1\}$

$$\#C = \frac{2^n}{\sum_{i=0}^1 \binom{n}{i}}$$

\Leftrightarrow

$$\#C = \frac{2^n}{\sum_{i=0}^1 \binom{n}{i}}$$

\Leftrightarrow

$$\#C = \frac{2^n}{\binom{n}{0} + \binom{n}{1}}$$

\Leftrightarrow

$$\#C = \frac{2^n}{1 + n}$$

Ahora hago la demostración:

Vuelta (\Leftarrow) suponiendo el antecedente:

Sea $r \in \mathbb{N}$ tal que $n = 2^r - 1 \wedge \#C = 2^{2^r-r-1}$ (existe por antecedente)

C es perfecto

\Leftrightarrow

$$\begin{aligned}
 \#C &= \frac{2^n}{1+n} \\
 \Leftrightarrow \{\text{Antecedente } n = 2^r - 1\} \\
 \#C &= \frac{2^{2^r-1}}{1+2^r-1} \\
 \Leftrightarrow \\
 \#C &= \frac{2^{2^r-1}}{2^r} \\
 \Leftrightarrow \\
 \#C &= 2^{2^r-1-r} \\
 \Leftrightarrow \{\text{Antecedente } \#C = 2^{2^r-r-1}\} \\
 \text{True}
 \end{aligned}$$

Ida (\Rightarrow) suponiendo el antecedente:

Por demostración auxiliar:

$$\begin{aligned}
 \#C &= \frac{2^n}{1+n} \\
 \Rightarrow \{\text{Los tamaños de conjuntos son números naturales}\} \\
 \frac{2^n}{1+n} &\in \mathbb{N}_0 \\
 \Rightarrow \{1+n \text{ divide a } 2^n\} \\
 (1+n) | 2^n \\
 \Rightarrow \{1+n \text{ es potencia de } 2\} \\
 \exists r \in \mathbb{N} : n = 2^r - 1 \\
 \Rightarrow \{\text{Demostración auxiliar: } \#C = \frac{2^n}{1+n}\} \\
 \exists r \in \mathbb{N} : n = 2^r - 1 \wedge \#C = \frac{2^{2^r-1}}{1+2^r-1} \\
 \Rightarrow \left\{ \frac{2^{2^r-1}}{1+2^r-1} = \frac{2^{2^r-1}}{2^r} = 2^{2^r-1-r} = 2^{2^r-r-1} \right\} \\
 \exists r \in \mathbb{N} : n = 2^r - 1 \wedge \#C = 2^{2^r-r-1}
 \end{aligned}$$

15)

miércoles, 11 de mayo de 2022 9:31

XV): Vimos arriba que un código binario perfecto que corrija un error debe tener longitud $2^r - 1$ y 2^{2^r-r-1} elementos. En el teórico vimos que para cada $r \geq 2$ existe un código lineal que corrige un error con esos parámetros. (los códigos de Hamming) ¿Existe alguno no lineal? La respuesta es si: Dado un $r \geq 2$, sea $n = 2^r - 1$, sea C el código:

$$C = \{(x, x \oplus w, x_1 \oplus \dots \oplus x_n \oplus (w_1 \vee \dots \vee w_n)) | x \in \mathbb{Z}_2^n, w \in \mathcal{H}_r\}$$

- a) Probar que la longitud de C es $2^{r+1} - 1$.
- b) Probar que C tiene $2^{2^{r+1}-(r+1)-1}$ elementos.
- c) Probar que $\delta = 3$ (por lo tanto por a) y b), C será perfecto). (damos una ayuda, pero lo pueden probar como quieran)
 - c1) Probar que $0 \in C$.
 - c2) Dar un $\alpha \in C$ con $d(\alpha, 0) = 3$. (esto y c1) prueba que $\delta \leq 3$.
 - c3) Probar que $\delta \geq 3$ probando que $d(\alpha, \beta) \geq 3 \forall \alpha, \beta \in C$, observando que si $\alpha = (x, x \oplus w, x_1 \oplus \dots \oplus x_n \oplus (w_1 \vee \dots \vee w_n))$ y $\beta = (y, y \oplus v, y_1 \oplus \dots \oplus y_n \oplus (v_1 \vee \dots \vee v_n))$, entonces se pueden hacer los siguientes casos:
 - c3i) $x \neq y, w = v, x_1 \oplus \dots \oplus x_n = y_1 \oplus \dots \oplus y_n$
 - c3ii) $x \neq y, w = v, x_1 \oplus \dots \oplus x_n \neq y_1 \oplus \dots \oplus y_n$.
 - c3iii) $x = y, w \neq v$.
 - c3iv) $x \neq y, w \neq v, x \oplus w = y \oplus v$.
 - c3v) $x \neq y, w \neq v, d(x \oplus w, y \oplus v) \geq 2$.
 - c3vi) $x \neq y, w \neq v, d(x \oplus w, y \oplus v) = 1$.
- d) Probar que C no es lineal.

Sea:

$$r \in \mathbb{N}_{\geq 2}$$

$$n = 2^r - 1$$

$$H \in \mathcal{H}_r$$

$$C = \left\{ \begin{bmatrix} x & x + w & \sum_{i=1}^n x_i + \sqrt[n]{w_i} \end{bmatrix} : x \in \{0,1\}^n, w \in H \right\}$$

a)

$$C \text{ es de longitud } 2^{r+1} - 1$$

Demostración:

$$\begin{aligned} & \text{Longitud de } C \\ &= \{x, x + w \in \{0,1\}^n\} \\ & 2n + 1 \\ &= \\ & 2 \cdot (2^r - 1) + 1 \end{aligned}$$

$$\begin{aligned}
&= \\
&= 2^{r+1} - 2 + 1 \\
&= 2^{r+1} - 1
\end{aligned}$$

b)

$$\#C = 2^{2^{r+1}-(r+1)-1}$$

Demostración:

$$\begin{aligned}
\#\{0,1\}^n &= 2^n \\
\#\mathcal{H}_r &= 2^{2^r-r-1}
\end{aligned}$$

$$\begin{aligned}
&\#C \\
&= \#\left\{ \left[\begin{matrix} x & x+w & \sum_{i=1}^n x_i + \bigvee_{i=1}^n w_i \end{matrix} \right] : x \in \{0,1\}^n, w \in H \right\} \\
&= \{\text{Por cada } x \text{ y } w \text{ se produce un elemento distinto}\} \\
&\quad \#\{0,1\}^n * \#H \\
&= 2^n * 2^{2^r-r-1} \\
&= \{n = 2^r - 1\} \\
&\quad 2^{2^r-1} * 2^{2^r-r-1} \\
&= 2^{2^r-1+2^r-r-1} \\
&= 2^{2*2^r-2-r} \\
&= 2^{2^{r+1}-(r+1)-1}
\end{aligned}$$

c)

$$\delta(C) = 3$$

Demostración:

Pruebo primero $\delta(C) \geq 3$ por contradicción (o sea, asumiendo $\delta(C) \leq 2$):

Sean:

$\alpha, \alpha' \in C$ con $\alpha \neq \alpha'$ tal que:

$$\begin{aligned}
\alpha &= [x \quad x+w \quad \sum_{i=1}^n x_i + \bigvee_{i=1}^n w_i] \text{ con } x \in \{0,1\}^n, w \in H \\
\alpha' &= [x' \quad x'+w' \quad \sum_{i=1}^n x'_i + \bigvee_{i=1}^n w'_i] \text{ con } x' \in \{0,1\}^n, w' \in H
\end{aligned}$$

En primer lugar:

$$\delta(C) \leq 2 \Rightarrow d(\alpha, \alpha') \leq 2$$

$$d(\alpha, \alpha') \leq 2 \Rightarrow d(x, x') \leq 2$$

$$d(\alpha, \alpha') \leq 2 \Rightarrow d(x + w, x' + w') \leq 2$$

$$\alpha \neq \alpha' \Rightarrow x \neq x' \vee w \neq w'$$

Divido en los siguientes casos:

$$w \neq w' \wedge x = x'$$

$$w = w' \wedge d(x, x') = 1$$

$$w = w' \wedge d(x, x') = 2$$

$$w \neq w' \wedge d(x, x') = 1$$

$$w \neq w' \wedge d(x, x') = 2$$

Llego a una contradicción en cada caso:

Caso $w \neq w' \wedge x = x'$:

$$d(\alpha, \alpha') \leq 2$$

\Rightarrow

$$d(x + w, x' + w') \leq 2$$

$$\Leftrightarrow \{x = x'\}$$

$$d(x + w, x + w') \leq 2$$

\Leftrightarrow

$$d(w, w') \leq 2$$

$$\Leftrightarrow \{w, w' \in H, \delta(H) = 3\}$$

False

Caso $w = w' \wedge d(x, x') = 1$:

$$d(\alpha, \alpha') \leq 2$$

\Leftrightarrow

$$d\left(\left[x \quad x + w \quad \sum_{i=1}^n x_i + \bigvee_{i=1}^n w_i\right], \left[x' \quad x' + w' \quad \sum_{i=1}^n x'_i + \bigvee_{i=1}^n w'_i\right]\right) \leq 2$$

$$\Leftrightarrow \{w = w', \text{ resta coordenada a coordenada}\}$$

$$\left\| \begin{bmatrix} x - x' & x - x' & \sum_{i=1}^n x_i - \sum_{i=1}^n x'_i \end{bmatrix} \right\| \leq 2$$

\Leftrightarrow

$$d(x, x') + d(x, x') + d\left(\sum_{i=1}^n x_i, \sum_{i=1}^n x'_i\right) \leq 2$$

$$\Leftrightarrow \{d(x, x') = 1, \text{ por ende tienen distinta cantidad de } 1\}$$

$$1 + 1 + 1 \leq 2$$

\Leftrightarrow

False

Caso $w = w' \wedge d(x, x') = 2$:

$$d(\alpha, \alpha') \leq 2$$

\Rightarrow

$$d(x, x') + d(x + w, x' + w') \leq 2$$

$\Leftrightarrow \{w = w'\}$

$$d(x, x') + d(x, x') \leq 2$$

$\Leftrightarrow \{d(x, x') = 2\}$

$$2 + 2 \leq 2$$

\Leftrightarrow

False

Caso $w \neq w' \wedge d(x, x') = 1$:

$$d(\alpha, \alpha') \leq 2$$

\Rightarrow

$$d(x, x') + d(x + w, x' + w') \leq 2$$

$\Leftrightarrow \{d(x, x') = 1\}$

$$d(x + w, x' + w') \leq 1$$

$\Leftrightarrow \{\text{Si o si hay al menos dos bits que son iguales en } x \text{ y } x' \text{ pero distintos en } w \text{ y } w'\}$

False

Caso $w \neq w' \wedge d(x, x') = 2$:

$$d(\alpha, \alpha') \leq 2$$

\Rightarrow

$$d(x, x') + d(x + w, x' + w') \leq 2$$

$\Leftrightarrow \{d(x, x') = 2\}$

$$d(x + w, x' + w') \leq 0$$

$\Leftrightarrow \{\text{Si o si hay al menos un bit que es igual en } x \text{ y } x' \text{ pero distinto en } w \text{ y } w'\}$

False

Ahora pruebo $\delta(C) \leq 3$:

$$\delta(C) \leq 3$$

\Leftarrow

$$\exists \alpha, \beta \in C : d(\alpha, \beta) \leq 3$$

\Leftarrow

$$\exists \alpha \in C : |\alpha| \leq 3$$

Si existe, doy un ejemplo:

Sea $\alpha \in C$ el elemento construido con $x = 1_n, w = 0_n$:

Pruebo que $|\alpha| \leq 3$:

$$\begin{aligned}
 |\alpha| &= \\
 &= \left\| \begin{bmatrix} 1_n & 1_n + 0_n & \sum_{i=1}^n 1 + \bigvee_{i=1}^n 0 \end{bmatrix} \right\| \\
 &= n + n + n \bmod 2 \\
 &\leq \{ \text{Si } n = 1 \text{ da justo 3, si } n \geq 2 \text{ claramente da mas de 3} \} \\
 &\quad 3
 \end{aligned}$$

d)

C es no lineal

Demostración por contradicción, o sea, supongo que C es lineal:

Sean:

$\alpha, \alpha' \in C$ con $\alpha \neq \alpha'$ talque:

$$\begin{aligned}
 \alpha &= [x \quad x + w \quad \sum_{i=1}^n x_i + \bigvee_{i=1}^n w_i] \text{ con } x \in \{0,1\}^n, w \in H \\
 \alpha' &= [x' \quad x' + w' \quad \sum_{i=1}^n x'_i + \bigvee_{i=1}^n w'_i] \text{ con } x' \in \{0,1\}^n, w' \in H
 \end{aligned}$$

C es lineal

$$\Rightarrow \alpha + \alpha' \in C$$

$$\Leftrightarrow \exists \beta \in C : \alpha + \alpha' = \beta$$

$$\Leftrightarrow \exists y \in \{0,1\}^n, v \in H : \alpha + \alpha' = \left[y \quad y + v \quad \sum_{i=1}^n y_i + \bigvee_{i=1}^n v_i \right]$$

Trabajo sin el \exists :

$$\begin{aligned}
 \alpha + \alpha' &= \left[y \quad y + v \quad \sum_{i=1}^n y_i + \bigvee_{i=1}^n v_i \right] \\
 \Leftrightarrow & \left[x + x' \quad x + w + x' + w' \quad \sum_{i=1}^n x_i + \bigvee_{i=1}^n w_i + \sum_{i=1}^n x'_i + \bigvee_{i=1}^n w'_i \right] = \left[y \quad y + v \quad \sum_{i=1}^n y_i + \bigvee_{i=1}^n v_i \right]
 \end{aligned}$$

$$x + x' = y \wedge x + w + x' + w' = y + v \wedge \sum_{i=1}^n (x_i + x'_i) + \bigvee_{i=1}^n w_i + \bigvee_{i=1}^n w'_i = \sum_{i=1}^n y_i + \bigvee_{i=1}^n v_i$$

Supongo los primeros dos términos , y trabajo con el tercero:

$$\begin{aligned} & \sum_{i=1}^n (x_i + x'_i) + \bigvee_{i=1}^n w_i + \bigvee_{i=1}^n w'_i = \sum_{i=1}^n y_i + \bigvee_{i=1}^n v_i \\ \Rightarrow & \{x + x' = y\} \\ & \bigvee_{i=1}^n w_i + \bigvee_{i=1}^n w'_i = \bigvee_{i=1}^n v_i \end{aligned}$$

Esto es falso para:

$$w = [1 \ 0 \ 0_{n-2}]$$

$$w' = [0 \ 1 \ 0_{n-2}]$$

$$v = [1 \ 1 \ 0_{n-2}]$$

$$\begin{aligned} & \bigvee_{i=1}^n w_i + \bigvee_{i=1}^n w'_i \\ = & \{\text{Solo hay unos tanto en } w \text{ como en } w'\} \\ & 1 + 1 \\ = & 0 \end{aligned}$$

$$\begin{aligned} & \bigvee_{i=1}^n v_i \\ = & \{\text{Hay unos en } v\} \\ & 1 \end{aligned}$$

XVI): En la isla de los caballeros y los picaros (recordar que los caballeros siempre dicen la verdad y los picaros siempre mienten) hay una mesa con 2048 llaves. Una de esas llaves abre la puerta de una habitación donde, del otro lado, hay oro, diamantes, la pocima de la salud eterna, y una máquina transdimensional que calcula $\chi(G)$ en $O(m)$. Las otras 2047 llaves destruirán la habitación si Ud. la usa para abrirla, además de infectarle a control remoto cualquier computadora en la cual Ud. alguna vez quiera hacer algo. (incluyendo las computadoras de los autos).

Para ayudarle a elegir la llave, 15 habitantes de la isla están dispuestos a responderle preguntas, cuyas respuestas sean "SI" o "NO". Una vez que uno de ellos ya haya respondido, no puede volver a preguntarle (a ese habitante). Cada uno sabe cuál es la llave, y estudiaron teoría de códigos, PERO NO saben quienes de entre esos 15 son caballeros y quienes picaros. Un decimosexto habitante de la isla le dice:

"Si soy caballero, entonces entre esas 15 personas hay a lo sumo un solo pícaro"
¿Cómo puede hacerle preguntas a las 15 personas para encontrar la llave?

En primer lugar, hay a los sumo un pícaro, ya que:

Sea:

φ = soy un caballero

ψ = hay a lo sumo un pícaro

Tengo que:

$$\varphi \Rightarrow (\varphi \Rightarrow \psi)$$

$$\neg\varphi \Rightarrow \neg(\varphi \Rightarrow \psi)$$

Pruebo ψ por casos:

Caso φ :

$$\begin{aligned} & \varphi \Rightarrow (\varphi \Rightarrow \psi) \\ \Rightarrow & \{\varphi\} \quad [\varphi]_1 \\ & \varphi \Rightarrow \psi \\ \Rightarrow & \{\varphi\} \quad \psi \\ & \psi \end{aligned}$$

Caso $\neg\varphi$ por contradicción:

$$\begin{aligned} & \neg\varphi \Rightarrow \neg(\varphi \Rightarrow \psi) \\ \Rightarrow & \neg(\varphi \Rightarrow \psi) \\ \Leftrightarrow & \neg(\text{False} \Rightarrow \text{False}) \\ \Leftrightarrow & \neg\text{True} \\ \Leftrightarrow & \text{False} \end{aligned}$$

Árbol de derivación:

$$\frac{\varphi \vee \neg\varphi \quad \frac{\begin{array}{c} [\varphi]_1 \quad [\neg\varphi]_1 \\ \varphi \Rightarrow (\varphi \Rightarrow \psi) \Rightarrow E \\ \varphi \Rightarrow \psi \end{array}}{\psi} \Rightarrow E \quad \frac{\begin{array}{c} [\psi]_2 \Rightarrow I \quad \neg\varphi \Rightarrow \neg(\varphi \Rightarrow \psi) \\ \varphi \Rightarrow \psi \quad \neg(\varphi \Rightarrow \psi) \\ \bot \end{array}}{\psi} \Rightarrow E}{\psi} \quad RAA_2 \quad \vee E_1$$

Ahora el ejercicio en si:

Quiero codificar un número entre 0 y 2047

Quiero codificarlo en 15 bits, y pasarlo por un canal que comete hasta un error.

$$n = 15$$

$$k = \dim(C) = \lceil \log_2(2048) \rceil = 11$$

Por ende la matriz de chequeo:

$$H \in \{0,1\}^{4 \times 15}$$

Necesito que sea capaz de corregir un error

$$\Rightarrow$$

$$\delta(C) \geq 3$$

$$\Rightarrow$$

No puede haber dos filas iguales en H

Sea:

$$H = [(1)_2^4 \quad (2)_2^4 \quad \cdots \quad (15)_2^4]$$