Resumen de teoremas para el final de Matemática Discreta II

Agustin Curto, agucurto95@gmail.com
Version 2017 Francisco Nievas

2016

Índice general

Capítulo 1

Parte C

1.1. La complejidad de EDMONS-KARP

Teorema: La complejidad de $\langle E-K\rangle$ con $n = |V|$ y $m = |E|$ es $\mathcal{O}(nm^2)$.

Prueba: Sean: $f_0, f_1, f_2, \ \ldots$ la sucesión de flujos creados por $\langle E-K\rangle$. Es decir, el paso k crea f_k.

Para cada k definimos funciones:

- $d_k(x) =$ "distancia" entre s y x en el paso k, en caso de existir, si no ∞.
- $b_k(x) =$ "distancia" entre x y t en el paso k, en caso de existir, si no ∞.

"Distancia": longitud del menor camino aumentante.

Sabemos que las distancias de $\langle E-K\rangle$ no disminuyen en pasos sucesivos, como esto será útil para esta demostración llamaremos $\circledast$ a la demostración de:

$$d_k(x) \leq d_{k+1}(x)$$

$$b_k(x) \leq b_{k+1}(x)$$

Llamemos *crítico* a un lado disponible en el paso k pero no disponible en el paso *k + 1*. Es decir, si xy es un lado $\Rightarrow$ xy se satura ó yx se vacía en el paso k.

1. Supongamos que al construir f_k el lado xy se vuelve crítico, el camino: $s \ldots$ x, y $\ldots$ t se usa para construir f_k.

$$\begin{aligned} d_k(t) &= d_k(x) + b_k(x) \\ &= d_k(x) + b_k(y) + 1 \end{aligned}$$

2. Para que xy pueda ser *crítico* nuevamente debe ser usado en la otra dirección. Sea l el paso posterior a k en el cual se usa el lado en la otra dirección, el camino $s \cdots$ y, x $\cdots$ t se usa para construir f_l.

$$\begin{aligned} d_l(t) &= d_l(x) + b_l(x) \\ &= d_l(y) + 1 + b_l(x) \end{aligned}$$

Entonces:

$$\text{De (1) y (2)} \Rightarrow \begin{cases} d_k(y) = d_k(x) + 1 & \star \\ d_l(x) = d_l(y) + 1 & \dagger \end{cases}$$

Luego:

$$\begin{aligned}
d_l(t) &= d_l(x) + b_l(x) & \\
&= d_l(y) + 1 + b_l(x) & \text{Por } \dagger \\
&\geq d_k(y) + 1 + b_k(x) & \text{Por } \circledast \\
&= d_k(x) + 1 + 1 + b_k(x) & \text{Por } \star \\
&= d_k(t) + 2 & \\
\therefore d_l(t) &\geq d_k(t) + 2 &
\end{aligned}$$

Por lo tanto cuando un lado se vuelve crítico recien puede volver a usarse cuando la distancia de s a t haya aumentado en por lo menos 2. Puede existir $\mathcal{O}(n/t)$ tales aumentos, es decir:

$$\#\text{ Veces que un lado puede volverse crítico } = \mathcal{O}(n).$$

$$\begin{aligned}
\therefore Complejidad(\langle E - K \rangle) &= (\#pasos) * Complejidad(1\ paso) \\
&= (\#\text{veces que un lado se vuelve crítico}) * (\#lados) * Complejidad(BFS) \\
&= \mathcal{O}(n) * \mathcal{O}(m) * \mathcal{O}(m) \\
&= \mathcal{O}(nm^2)
\end{aligned}$$

Q.E.D.

1.2. Las distancias de EDMONS-KARP no disminuyen en pasos sucesivos

Teorema: Sean: $f_0, f_1, f_2, \ \ldots$ la sucesión de flujos creados por $\langle E - K \rangle$. Es decir, el paso k crea f_k.

Para cada k definimos funciones:

- $d_k(x) =$ "distancia" entre s y x en el paso k en caso de existir, si no ∞.
- $b_k(x) =$ "distancia" entre x y t en el paso k en caso de existir, si no ∞.

"Distancia": longitud del menor camino aumentante.

Queremos probar que:

1. $d_k(x) \leq d_{k+1}(x)$
2. $b_k(x) \leq b_{k+1}(x)$

Prueba: Lo probaremos por inducción y solo para d_k ya que para b_k la prueba es análoga.

$$\text{HI: H(i)} \ = \{\forall_z : d_{k+1}(z) \leq i, \ \text{vale } d_k(z) \leq d_{k+1}(z)\}$$

1. Caso Base: $\boxed{\text{i} = 0}$ $\qquad H(0) = \{\forall_z : d_{k+1}(z) \leq 0, \ \text{vale } d_k(z) \leq d_{k+1}(z)\}$
   Pero $d_{k+1}(z) \leq 0 \Rightarrow z = s$, entonces:

$$\begin{array}{rcl} d_k(z) & = & d_k(s) \\ & = & 0 \\ & \leq & d_{k+1}(s) \\ & = & d_{k+1}(z) \\ \therefore d_k(z) & \leq & d_{k+1}(z) \end{array}$$

2. Caso Inductivo: Supongamos ahora que vale H(i), veamos que vale H(i + 1).
   Sea z con $d_{k+1}(z) \leq i$+1, si $d_{k+1}(z) \leq i$ vale H(i) para z.

$$\therefore \ d_k(z) \leq d_{k+1}(z)$$

   Supongamos que $\boxed{d_{k+1}(z) = i + 1}$ $\circledast$
   Entonces existe un camino aumentante, relativo a f_k, de la forma: $s = z_0, \ \ldots \ z_i, \ z_{i+1} = z$.
   Sea $\boxed{x = z_i}$

   - Caso 1: Existe algun camino aumentante, relativo a f_{k-1}, de la forma $s, \ \ldots \ x, \ z$.

$$\therefore d_k(z) \leq d_k(x) + 1$$

     Pues al haber un camino $\underbrace{s, \ \ldots \ x,}_{d_k(x)} \ z$, llamemosle A, de longitud $d_k(x) + 1$ entre s y z, sabemos que el mínimo de todos los caminos de s a z serán $\leq$ A.

- <u>Caso 2:</u> No existe un camino aumentante, relativo a f_{k-1}, pero si existe un camino aumentante relativo a f_k. Por lo tanto el lado xz no esta "disponible" en el paso k, ya que xz está saturado, o bien zx está vacío relativo a f_{k-1}. Es decir:

  1) $f_{k-1}(\overrightarrow{xz}) = Cap(\overrightarrow{xz})$ pero $f_k(\overrightarrow{xz}) < Cap(\overrightarrow{xz}),\ f_k$ devuelve flujo por $\overrightarrow{xz}$ ó
  2) $f_{k-1}(\overrightarrow{zx}) = 0$ pero $f_k(\overrightarrow{zx}) > 0,\ f_k$ manda flujo por $\overrightarrow{zx}$.

  Para construir f_k usamos un camino de la forma $s,\ \dots\ z,\ x$.
  Como $\langle E-K\rangle$ funciona con BFS, ese camino usado para construir f_k debe ser de longitud mínima. Es decir:

$$\begin{array}{rcl} d_k(x) & = & d_k(z)+1 \\ \Rightarrow d_k(z) & = & d_k(x)-1 \\ & \leq & d_k(x)+1 \end{array}$$

<u>Conclusión:</u> En cualquiera de los dos casos tenemos:

$$\boxed{d_k(z) \leq d_k(x)+1} \quad (1)$$

Ahora bien:

$$\begin{array}{rcll} d_{k+1}(x) & = & d_{k+1}(z_i) & \\ & = & i & (2) \\ & \Rightarrow & \text{H}(i) \text{ vale para x.} & \\ \therefore\ d_k(x) & \leq & d_{k+1}(x) & (3) \end{array}$$

Por lo tanto:

$$\begin{array}{rcll} d_k(z) & \leq & d_k(x)+1 & \text{Por (1)} \\ & \leq & d_{k+1}(x)+1 & \text{Por (3)} \\ & = & i+1 & \text{Por (2)} \\ & = & d_{k+1}(z) & \text{Por } \circledast \\ & \Rightarrow & \text{H(i + 1) vale.} & \end{array}$$

Q.E.D.

1.3. La complejidad de DINIC

Teorema: La complejidad del algoritmo de Dinic es $\mathcal{O}(n^2m)$.

Prueba: Como Dinic es un algoritmo que trabaja con networks auxiliares y vimos que la distancia entre s y t en networks auxiliares consecutivos aumenta y puede ir a lo sumo entre 1 y $n-1$ entonces hay a lo sumo $\mathcal{O}(n)$ networks auxiliares.

$$\text{Complejidad(Dinic)} = \mathcal{O}(n) * \text{Complejidad(Paso Bloqueante de Dinic)}$$

Para probar que la complejidad de Dinic es $\mathcal{O}(n^2m)$ debemos probar que la complejidad del paso bloqueante es $\mathcal{O}(nm)$.

Sean:

- A = Avanzar()
- R = Retroceder()
- I = IncrementarFlujo() + Inicialización()

Una corrida de Dinic luce como:

$$\text{AA} \ldots \text{AIAAARA} \ldots \text{AIAARAAARR} \ldots \text{IA} \ldots$$

Dividamos la corrida en subpalabras del tipo:

$$\left.\begin{array}{l} \underbrace{AA\ \ldots\ A}_{Todas\ A's}R \\ \underbrace{AA\ \ldots\ A}_{Todas\ A's}I \end{array}\right\} \text{Sea X = I o R}$$

Nota: el número de A's puede ser 0.

Debemos determinar:

1. **Cantidad de subpalabras**

$$\left.\begin{array}{ll} X = R: & \text{borramos un lado} \\ X = I\ : & \text{borramos al menos un lado} \end{array}\right\} \Rightarrow \text{ Cada X borra al menos un lado}$$

$$\therefore \text{ hay } \leq m \text{ palabras de la forma } A \ldots AX$$

2. **Complejidad de cada subpalabra**

Recordemos que:

$$\text{A:} \left[\begin{array}{l} P[i+1] = \text{algún elemento de } \Gamma^+(P[i]) \\ i = i+1 \end{array} \right.$$

$$\Rightarrow \text{ A es } \mathcal{O}(1)$$

$$\text{R:} \left[\begin{array}{l} \text{Borrar } P[i-1]P[i] \text{ del NA} \\ i = i-1 \end{array} \right.$$

$$\Rightarrow \text{ R es } \mathcal{O}(1)$$

$$\text{I:} \left[\text{ Recorre un camino de longitud } \leq n \right.$$

$$\Rightarrow \text{ I es } \mathcal{O}(n)$$

Luego:

$$\begin{aligned} Complejidad(X) &= \mathcal{O}(n) \\ \therefore Complejidad(\underbrace{A\ \ldots\ A}_{j\ veces}X) &= \mathcal{O}(j) + \mathcal{O}(n) \end{aligned}$$

Como cada Avanzar() mueve el pivote un nivel más cerca de t entonces hay a lo sumo n Avanzar() antes de un R o un I $\Rightarrow j \leq$ n.

$$\therefore\ Complejidad(A \ldots AX) = \mathcal{O}(n) + \mathcal{O}(n) = \mathcal{O}(n)$$

De (1) y (2):

$$\begin{aligned} \text{Complejidad(Paso Bloqueante de Dinic)} &= \#(A \ldots AX) * \text{Complejidad}(A \ldots AX) \\ &= m * \mathcal{O}(n) \\ &= \mathcal{O}(m * n) \end{aligned}$$

Q.E.D.

1.4. La complejidad de WAVE

Teorema: La complejidad del algoritmo de Wave es $\mathcal{O}(n^3)$.

Prueba: Como Wave es un algoritmo que trabaja con networks auxiliares y vimos que la distancia entre s y t en networks auxiliares consecutivos aumenta y puede ir a lo sumo entre 1 y $n-1$ entonces hay a lo sumo $\mathcal{O}(n)$ networks auxiliares.

$$\text{Complejidad(Wave)} = \mathcal{O}(n) * \text{Complejidad(Paso Bloqueante de Wave)}$$

Para probar que la complejidad de Wave es $\mathcal{O}(n^3)$ debemos probar que complejidad del paso bloqueante es $\mathcal{O}(n^2)$. El paso bloqueante de Wave consiste en una serie de:

- Olas hacia adelante: Sucesión de **fordwardbalance** (FB)
- Olas hacia atrás: Sucesión de **backwardbalance** (BB)

Cada FB y BB es una sucesión de "**buscar vecinos**" y "**procesar**" el lado resultante. Estos "procesamientos" son complicados pero $\mathcal{O}(1)$.

$$\therefore\ \text{Complejidad(Paso Bloqueante de Wave)} = \#\ \text{"procesamientos de lado"}$$

Los "procesamientos" de lados los podemos dividir en dos categorías:

1. Aquellos procesamientos que saturan o vacian el lado. Denotaremos "T" al número de estos procesamientos.
2. Aquellos procesamientos que no saturan ni vacian el lado. Denotaremos "Q" al número de estos procesamientos.

Por lo tanto queremos acotar $T + Q$.

Complejidad de T:

- ¿Puede un lado $\overrightarrow{xy}$ saturado volver a saturarse?

  Para poder volver a <u>saturarse</u> primero tiene que vaciarse aunque sea un poco, es decir, antes de poder volver a saturarlo y debe devolver flujo a x, pero para que en Wave y le devuelva flujo a x debe ocurrir que y esté bloqueado (porque BB(y) solo se ejecuta si y está bloqueado), pero si y está bloqueado x no puede mandarle flujo nunca más.

  $$\therefore \overrightarrow{xy} \text{ no puede resaturarse}$$

  <u>Conclusión 1:</u> Los lados se saturan solo una vez.

- ¿Puede un lado $\overrightarrow{xy}$ vaciado completamente volver a vaciarse?

  Para poder volver a <u>vaciarse</u> como está vacío completamente, primero hay que mandar flujo, pero si lo vacié, y está bloqueado por lo que x no puede mandar flujo.

  $$\therefore \overrightarrow{xy} \text{ no puede volver a vaciarse}$$

  <u>Conclusión 2:</u> Los lados se vacían completamente a lo sumo una vez.

  Las conclusiones (1) y (2) implican que $\boxed{T \leq 2\,m}$

Complejidad de Q:

En cada FB a lo sumo un lado no se satura y en cada BB a lo sumo un lado no se vacía completamente.

$$\therefore Q \leq \#\text{ Total de FB's y BB's}$$

- \# FB's en cada ola hacia adelante es $\leq n$ (un FB por vértice)
- \# BB's en cada ola hacia atrás es $\leq n$ (un BB por vértice)

$$\therefore \text{Total de FB's y BB's} \leq 2n\ \#\text{Total de ciclos de ``ola adelante} - \text{ola hacia atrás''}$$

Ahora bien, en cada ola hacia adelante, pueden o no, bloquearse algunos vértices. Si no se bloquea ningún vértice, entonces todos los vértices ($\neq s,\ t$) quedan balaceados por lo que estamos en la última ola. Luego en toda ola que no sea la última se bloquea al menos un vértice ($\neq s,\ t$).

$$\therefore\ \#\text{ Total de ciclos es} \leq (n-2)+1 = n-1$$

$$\Rightarrow \boxed{Q\ \leq 2n\,(n-1) = \mathcal{O}(n^2)}$$

$$\begin{aligned}\therefore\ T + Q &\leq 2m + \mathcal{O}(n^2)\\ &= \mathcal{O}(m) + \mathcal{O}(n^2)\\ &= \mathcal{O}(n^2)\end{aligned}$$

Q.E.D.

Capítulo 2

Parte N

2.1. 3-COLOR es NP-Completo

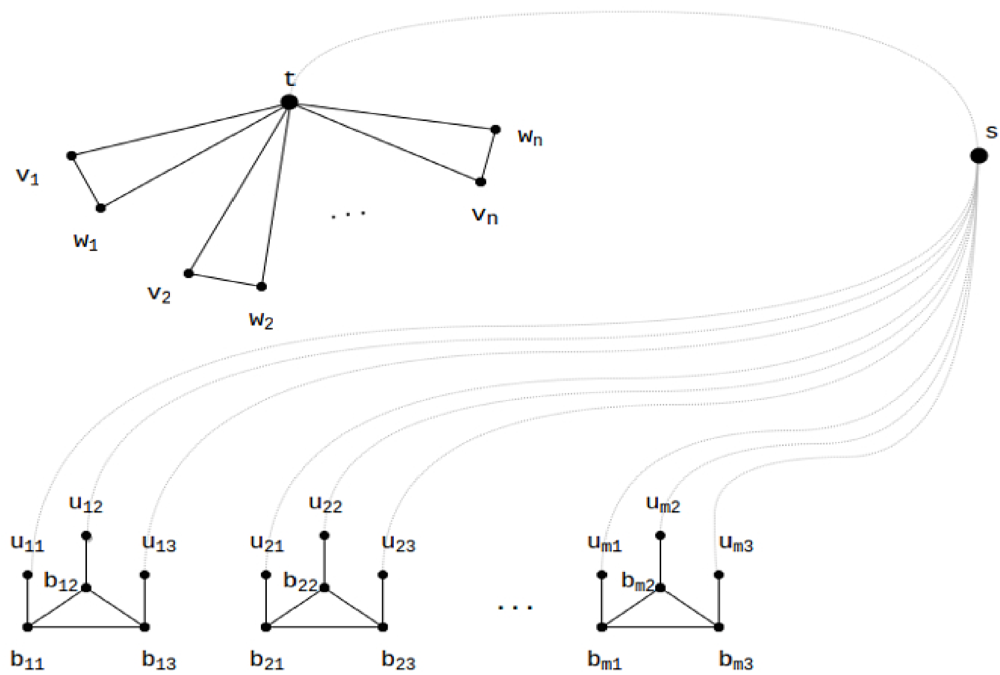
Teorema: 3-COLOR es NP-Completo
Prueba: Veremos que 3-SAT $\leq_p$ 3-COLOR, es decir, dado B en CNF con 3 literales por disyunción, debemos crear un grafo G, tal que:

$$\text{B es satisfacible } \Leftrightarrow \chi(G) \leq 3$$

Sea:

$$\begin{aligned} B &= D_1, D_2 \dots D_m \text{ con variables } \{x_1 \dots x_m\} \\ D_i &= l_{i,1} \vee l_{i,2} \vee l_{i,3} \end{aligned}$$

Nuestro G será un $G_1 = (V, E \cup F)$, es decir, $G = (V, E)$ con lados extra F, determinados según B. Este es G:



Construcción del grafo:

- Dado un literal l, definimos:

$$\psi(l) = \begin{cases} v_k & \text{si } l = x_k \\ w_k & \text{si } l = \overline{x_k} \end{cases}$$

- Vértices:

$$V(G) = \{s,t\} \cup \{v_1 \dots v_n, w_1 \dots w_n\} \cup \{\mu_{i,1}, \mu_{i,2}, \mu_{i,3}, b_{i,1}, b_{i,2}, b_{i,3}\}_{i=1}^m$$

- Aristas:

$$E(G) = \{st\} \cup \{tv_i, tw_i, v_iw_i\}_{i=1}^n \cup \{s\mu_{i,j}\}_{i=1,\ j=1,2,3}^m \cup F \cup \{b_{i,1}b_{i,2}, b_{i,1}b_{i,3}, b_{i,2}b_{i,3}, b_{i,1}\mu_{i,1}, b_{i,2}, \mu_{i,2}, b_{i,3}\mu_{i,3}\}_{i=1}^m$$

Donde:

$$F = \{\mu_{i,j}\psi(l_{i,j})\}_{i=1,\ j=1,2,3}^m$$

$$\text{G tiene: } \left\{ \begin{array}{l} 2+2n+6m \text{ vértices} \\ 1+3n+3m+3m+6m \text{ aristas} \end{array} \right\} \Rightarrow \text{ G es polinomial}$$

$\boxed{\Leftarrow}$

Suponemos $\chi(G) \leq 3$ y construiremos un $\overrightarrow{b}$ tal que $B(\overrightarrow{b}) = 1$. Como G tiene triángulos, si $\chi(G) \leq 3$, entonces debe ser $\chi(G) = 3$. Por lo tanto, existe un coloreo C de G con 3 colores.

Definición:

$$b_k = \left\{ \begin{array}{l} 1 \text{ si } C(v_k) = C(s) \\ 0 \text{ si } C(v_k) \neq C(s) \end{array} \right.$$

Para probar $B(\overrightarrow{b}) = 1$ debemos probar que $D_i(\overrightarrow{b}) = l_{i,1} \vee l_{i,2} \vee l_{i,3} = 1\ \forall\ i$.

Sea $i \in \{1, 2, \dots m\}$, como $\{b_{i,1}, b_{i,2}, b_{i,3}\}$ es un triángulo, entonces deben aparecer los 3 colores. Es decir, $\exists j : C(b_{i,j}) = C(t)$

Luego:

$$\left. \begin{array}{l} (1)\ \mu_{i,j}b_{i,j} \in E(G) \Rightarrow C(\mu_{i,j}) \neq C(b_{i,j}) = C(t) \\ (2)\ \mu_{i,j}s \in E(G) \Rightarrow C(\mu_{i,j}) \neq C(s) \\ (3)\ st \in E(G) \Rightarrow C(s) \neq C(t) \end{array} \right\} \Rightarrow C(\mu_{i,j}) = \text{ TERCER COLOR}$$

Por otro lado:

$$\left. \begin{array}{l} (1)\ \mu_{i,j}\psi(l_{i,j}) \in E(G) \Rightarrow C(\psi(l_{i,j})) \neq \text{ TERCER COLOR} \\ (2)\ \psi(l_{i,j})t \in E(G) \Rightarrow C(\psi(l_{i,j}) \neq C(t) \end{array} \right\} \Rightarrow C(\psi(l_{i,j}) = C(s)$$

- <u>Caso (1):</u>

$$l_{i,j} = x_k \Rightarrow \left\{ \begin{array}{l} l_{i,j}(\overrightarrow{b}) = b_k \\ \psi(l_{i,j}) = v_k \end{array} \right.$$

Entonces:

$$\begin{array}{rcl} C(v_k) & = & C(s) \Rightarrow b_k = 1 \\ \therefore l_{i,j}(\overrightarrow{b}) & = & 1 \Rightarrow D_i(\overrightarrow{b}) = 1 \end{array}$$

- <u>Caso (2):</u>

$$l_{i,j} = \overline{x_k} \Rightarrow \left\{ \begin{array}{l} l_{i,j}(\overrightarrow{b}) = \overline{b_k} \\ \psi(l_{i,j}) = w_k \end{array} \right.$$

Entonces:

$$\left. \begin{array}{l} C(w_k) = C(s) \\ v_kw_k \in E(G) \end{array} \right\} \Rightarrow C(v_k) \neq C(s) \therefore b_k = 0$$

$$\therefore \overline{b_k} = 1 \Rightarrow D_i(\overrightarrow{b}) = 1$$

$\boxed{\Rightarrow}$

Acá asumimos que $\exists \overrightarrow{b} : B(\overrightarrow{b}) = 1 \Rightarrow \forall i,\ D(\overrightarrow{b}) = 1 \Rightarrow \boxed{\forall i\ \exists j : l_{i,j}(\overrightarrow{b}) = 1}$ $(\star)$. Debemos construir un coloreo propio con 3 colores.

Definimos:

$$\left.\begin{array}{l} C(s) = \text{ BLANCO} \\ C(t) = \text{ AZUL} \end{array}\right\} \Rightarrow st \text{ No Crea Problemas (NCP)}$$

$$C(v_k) = \left\{\begin{array}{ll} \text{BLANCO} & \text{si } b_k = 1 \\ \text{NEGRO} & \text{si } b_k = 0 \end{array}\right\} \Rightarrow \underbrace{v_k}_{B\ o\ N}\ \underbrace{w_k}_{B\ o\ N} \text{ NCP}$$

$$C(w_k) = \left\{\begin{array}{ll} \text{NEGRO} & \text{si } b_k = 1 \\ \text{BLANCO} & \text{si } b_k = 0 \end{array}\right\} \Rightarrow \underbrace{v_k}_{BoN}\underbrace{t}_{A} \text{ y } \underbrace{w_k}_{BoN}\underbrace{t}_{A} \text{ NCP}$$

Falta colorear las garras. Dado i, tomemos el j de $(\star)$ y definimos:

$$C(\mu_{i,r}) = \left\{\begin{array}{ll} \text{NEGRO} & \text{si } r = j \\ \text{AZUL} & \text{si } r \neq j \end{array}\right\} \Rightarrow \underbrace{\mu_{i,r}}_{N\ o\ A}\ \underbrace{s}_{B} \text{ NCP } \forall r$$

- <u>Caso $r \neq j$:</u> $\underbrace{\mu_{i,r}}_{A}\underbrace{\psi(l_{i,r})}_{B\ o\ N} \Rightarrow \text{NCP}$
- <u>Caso $r = j$:</u>
  - Caso (1):

$$l_{i,j} = x_k \Rightarrow \left\{\begin{array}{l} l_{i,j}(\overrightarrow{b}) = b_k = 1 \Rightarrow C(v_k) = \text{ BLANCO} \\ \psi(l_{i,j}) = v_k \Rightarrow \mu_{i,j}\psi(l_{i,j}) = \underbrace{\mu_{i,j}}_{N}\ \underbrace{v_k}_{B} \text{ NCP} \end{array}\right.$$

  - Caso (2):

$$l_{i,j} = \overline{x_k} \Rightarrow \left\{\begin{array}{l} l_{i,j}(\overrightarrow{b}) = \overline{b_k} = 1 \Rightarrow b_k = 0 \Rightarrow C(w_k) = \text{ BLANCO} \\ \psi(l_{i,j}) = w_k \Rightarrow \mu_{i,j}\psi(l_{i,j}) = \underbrace{\mu_{i,j}}_{N}\ \underbrace{w_k}_{B} \text{ NCP} \end{array}\right.$$

Quedan las bases:

$$C(b_{i,r}) = \left\{\begin{array}{ll} \text{AZUL} & \text{si } r = j \\ \text{BLANCO o NEGRO} & \text{si } r \neq j \\ \text{NEGRO o BLANCO} & \text{si } r \neq j \end{array}\right.$$

Es decir, le damos el color AZUL al $b_{i,j}$ y los otros dos los coloreamos uno NEGRO y uno BLANCO, de modo que:

- $\{b_{i,1}, b_{i,2}, b_{i,3}\}$ NCP
- $\underbrace{b_{i,j}}_{A}\ \underbrace{\mu_{i,j}}_{N}$ NCP
- $\underbrace{b_{i,r}}_{B\ o\ N}\ \underbrace{\mu_{i,r}}_{A}$ NCP

Q.E.D.

Capítulo 3

Parte G

3.1. Greedy no empeora

Teorema: Sea G es un grafo, y C un coloreo propio cualquiera de G con r colores y V_1 son los vertices coloreados con 1, V_2 los coloreados con 2, ..., V_r los coloreados con r, entonces si se ordenan los vertices poniendo primero los vertices de V_{j_1} luego los de V_{j_2} , ..., V_{j_r}, donde $\{j_1, j_2, \ldots, j_r\}$ es un ordenamiento arbitrario de los colores, entonces Greedy colorea a G con el nuevo orden con a lo sumo r colores.

Prueba: Probaremos por inducción en r

Caso Base: $\boxed{\text{r} = 0}$ No hay nada que probar. Ya que si G tiene un solo color nos dice que G no tiene lados, por lo tanto Greedy en cualquier orden colorea con un color.

Caso Inductivo: Supongamos ahora que vale para r, veamos que vale para $r + 1$.

Sea:

- W = Vértices de colores $j_1, j_2, \ldots, j_k$
- U = Vértices de colores j_{k+1}

Sea: $H = G[W]$

$C\mid_H$ Colorea H con k colores

Luego, por HI Greedy(H) con el orden de los vértices "*por colores*" obtendrá $l \leq k$ colores. Entonces, Greedy(H) va a colorear los vértices de H con esos l colores. Veamos cuantos colores extras se requieren para colorear U.

Si: $x \in U$

A) Si $\exists\ j \leq l$ tal que x no es vecino de ningún vértice de color j. Greedy colorea x con j (o menos) $\therefore$ no requiere un color extra.

B) Si no, Greedy le dará un color extra $l + 1$. Greedy no se puede ver forzado a darle el color $l + 2$ ya que para que eso ocurra necesariamente x tiene que ser vecino de otro vértice de color $l + 1$. Los únicos que tienen color $l + 1$ son los vértices de U. Pero no hay lados entre los vértices de U pues C los colorea a todos con el color $j + 1$ y C es propio.

$\therefore$ Greedy usa a lo sumo $l + 1 \leq k + 1$ colores.

Q.E.D.

3.2. 2-COLOR es polinomial

Teorema: 2-Color es polinomial, es decir, existe un algoritmo polinomial que lo resuelve.
Prueba: Consideremos el siguiente algoritmo con entrada $G = (V, E)$ con $n = |V|$ y $m = |E|$.

Algoritmo:
Por cada COMPONENTE CONEXA de G:

1. Elegir x
2. Correr BFS(x)
3. Colorear vertices con: $C(v) = Nivel_{BFS(x)}(v) \; mod \; 2$
4. Chequear que el coloreo de (3) sea propio. Si lo es, RETURN "SI", si no lo es, RETURN "NO".

Veamos ahora que el algoritmo es correcto y que su complejidad es polinomial.

- Complejidad Polinomial:
  - (1) es $\mathcal{O}(1)$.
  - (2) + (3) es $\mathcal{O}(m + a)$ recorrer todas las aristas, donde $a = \#$vertices aislados.
  - (4) es $\mathcal{O}(m)$ ya que chequear que un coloreo es propio, es recorrer todos los lados comprobando que los vértices tengan distintos colores.
- Correctitud:
  - Cuando dice "SI": Obviamente dice "SI", solo si el coloreo es propio y si el coloreo es propio, $\chi(G) \leq 2$.
  - Cuando dice "NO": Tenemos que ver que cuando dice "NO" ningún coloreo propio con 2 colores existe.

    Por otro teorema visto en clase, si existe un ciclo impar $C_{2k+1} \subseteq G \Rightarrow \chi(G) \geq 3$. Por lo que bastaría ver que si el algoritmo dice "NO", entonces exista un ciclo impar en G.

    Si el algoritmo dice "NO"$\Rightarrow \exists y, z : yz \in E(G)$ y $C(y) = C(z) \Rightarrow Nivel_{BFS(x)}(y) \equiv Nivel_{BFS(x)}(z)(2)$.

    Sea:
    - $x \dots y$ un camino en BFS(x), entre x e y.
    - $x \dots z$ un camino en BFS(x), entre x y z.
    - $s = Nivel_{BFS(x)}(y)$ y $t = Nivel_{BFS(x)}(z)$ $(s \equiv t \; (2))$.
    - w el vértice de mayor nivel tal que: $x \dots w$ sea prefijo común en $x \dots y$ y $x \dots z$. Es decir:

$$\begin{array}{cc} x \dots w & \dots y \\ \underbrace{x \dots w}_{\text{Camino Igual}} & \underbrace{\dots z}_{\text{Camino distinto}} \end{array}$$

- $r = Nivel_{BFS(x)}(w)$

Considramos el ciclo: $w \ldots \overbrace{yz}^{Por\ Hipotesis} \ldots w$

$$\begin{aligned} \text{Longitud del ciclo} &= (s-r)+1+(t-r) \\ &= (s+t)-2r+1 \\ &\equiv 0+0+1 = 1\ (2) \end{aligned}$$

$\therefore$ Es un ciclo impar

Q.E.D.

3.3. Teorema Max-Flow Min-Cut

<u>Teorema:</u>

a) Si f es flujo y S es corte $\Rightarrow$ V(f) $\leq$ Cap(S).

b) Si V(f) = Cap(S) $\Rightarrow$ f es maximal y S es minimal.

c) Si f es maximal $\Rightarrow$ $\exists$ S con V(f) = Cap(S).

<u>Prueba:</u> Denotemos ($\star$), a la demostración de:

$$\boxed{V(f) = f(S,\overline{S}) - f(\overline{S},S)}$$

a) $\boldsymbol{f}$ es flujo y S es corte $\Rightarrow$ V($\boldsymbol{f}$) $\leq$ Cap(S).

$$\begin{aligned} V(f) &\overset{Por\ (\star)}{=} f(S,\overline{S}) \underbrace{- \underbrace{f(\overline{S},S)}_{\geq 0}}_{\leq 0} \\ &\leq f(S,\overline{S}) \\ &\leq Cap(S,\overline{S}) \\ &= Cap(S) \end{aligned}$$

b) V($\boldsymbol{f}$) = Cap(S) $\Rightarrow$ $\boldsymbol{f}$ es maximal y S es minimal.

Supongamos que V(f) = Cap(S). Sea g un flujo cualquiera y T un corte cualquiera.

- $V(g) \overset{Por\ a)}{\leq} Cap(S) = V(f) \Rightarrow$ f es maximal.
- $Cap(T) \overset{Por\ a)}{\geq} V(f) = Cap(S) \Rightarrow$ S es minimal.

c) *f* es maximal $\Rightarrow \exists$ S con V(*f*) = Cap(S).

Sea $S = \{s\} \cup \{x : \exists$ camino aumentante, realtivo a *f*, entre s y $x\}$

$\boxed{¿t \in S?}$

<u>Si t estuviera en S:</u> existiría un camino aumentante entre s y t.
Por el teorema del camino aumentante podemos construir un flujo g tal que:

$$\begin{aligned} V(g) &= V(f) + \epsilon \text{ para algun } \epsilon > 0 \\ \Rightarrow V(g) &> V(f) \text{ Absurdo! pues f es maximal} \\ \therefore t &\notin S \Rightarrow \text{ S es corte.} \end{aligned}$$

Solo resta ver que: $V(f) = Cap(S)$

Por $(\star)$: $V(f) = \underbrace{f(S, \overline{S})}_{(1)} - \underbrace{f(\overline{S}, S)}_{(2)}$

Analicemos (1) y (2)

(1) $f(S, \overline{S}) = \sum_{\substack{x \in S \\ y \in \overline{S} \\ xy \in E}} f(\overrightarrow{xy})$

$x \in S \Rightarrow \exists$ camino aumentante $s \ldots x$.

$y \in \overline{S} \Rightarrow \nexists$ camino aumentante entre s e y.

En particular $s \ldots x \ldots y$ no es camino aumentante, por lo que no puede darse que:

$$f(\overrightarrow{xy}) < Cap(\overrightarrow{xy})$$

$\Rightarrow f(\overrightarrow{xy}) = Cap(\overrightarrow{xy}) \qquad \forall x \in S, \forall y \in \overline{S} : \overrightarrow{xy} \in E.$

$\Rightarrow \boxed{f(S, \overline{S})} = \sum_{\substack{x \in S \\ y \in \overline{S} \\ xy \in E}} f(\overrightarrow{xy}) = \sum_{\substack{x \in S \\ y \in \overline{S} \\ xy \in E}} Cap(\overrightarrow{xy}) = Cap(S, \overline{S}) = \boxed{Cap(S)}$

(2) $f(\overline{S}, S) = \sum_{\substack{x \in \overline{S} \\ y \in S \\ xy \in E}} f(\overrightarrow{xy})$

$x \in \overline{S} \Rightarrow \nexists$ camino aumentante entre s y x.

$y \in S \Rightarrow \exists$ camino aumentante $s \ldots y$.

En particular $s \ldots y \ldots x$ no es camino aumentante $\Rightarrow f(\overrightarrow{xy}) = 0 \ \forall x \in \overline{S}, \forall y \in S : \overrightarrow{xy} \in E.$

$$\therefore \boxed{f(\overline{S}, S) = 0}$$

Luego de (1) y (2):

$$\begin{aligned} V(f) &= f(S, \overline{S}) - f(\overline{S}, S) \\ &= Cap(S) - 0 \\ &= Cap(S) \end{aligned}$$

Q.E.D.

3.4. Teorema de Hall

Teorema: Sea G = (X ∪ Y, E) grafo bipartito, entonces ∃ matching completo de:

$$X \text{ a } Y \Leftrightarrow |S| \leq |\Gamma(S)| \; \forall \text{ S} \subseteq X$$

Prueba:

⇒) Si M es matching completo de X a Y entonces observemos que M induce una función inyectiva de X a Y.

$$f(x) = \text{ unico } y : xy \in E(M)$$

1. Si $S \subseteq X \Rightarrow |S| \leq |f(S)|$

2. Además por definición de f:

$$\begin{aligned} f(x) \in \Gamma(x) &\Rightarrow f(S) \subseteq \Gamma(S) \\ &\therefore |f(S)| \leq |\Gamma(S)| \end{aligned}$$

De (1) y (2) $\Rightarrow |S| \leq |\Gamma(S)|$.

⇐) Supongamos que no es cierto, entonces G es bipartito con $|S| \leq |\Gamma(S)| \; \forall S \subseteq X$ pero no tiene matching completo de X a Y.

Cuando corremos el algoritmo quedan filas sin matchear (las de s).

Sean:

- $S_0 = \{\text{filas sin matchear}\}$.
- $T_1 = \Gamma(S_0)$, es decir, las columnas etiquetadas por las filas de S_0. Todas las columnas de T_1 están matcheadas, pues si no, se podría agregar alguna alguna fila de S_0 al matching.
- $S_1 = \{\text{filas etiquetadas por las columnas de } T_1\}$.
- $T_2 = \Gamma(S_1) - T_1$, es decir, columnas etiquetadas por las filas de S_1.

En general:

- $S_i = \{\text{filas matcheadas con } T_i\}$.
- $T_{i+1} = \Gamma(S_i) - (T_1 \cup T_2 \cup \dots T_i)$.

Como el algoritmo para sin hallar matching, entonces $\forall i \; T_i \neq \emptyset$, produce un S_i (i.e $S_i \neq \emptyset$).
$\therefore$ La única forma de parar es en un k, tal que $T_{k+1} = \emptyset$.

Observaciones:

1. $|S_i| = |T_i|$, pues S_i son las filas matcheadas con T_i.
2. Por construcción, los S_i y T_i son todos distintos.
3. $\Gamma(S_0 \cup S_1 \cup \dots S_j) = T_1 \cup T_2 \cup \dots T_{j+1}$
   Por inducción en j:
   - Caso Base: $j = 0$ vale, ya que $\Gamma(S_0) = T_1$
   - Caso Inductivo: Supongamos que vale para j, veamos para $j+1$.

$$\begin{aligned}
T_1 \cup T_2 \dots T_{j+2} &= T_1 \cup T_2 \cup \dots T_{j+1} \cup \underbrace{(\Gamma(S_{j+1}) - (T_1 \cup T_2 \dots T_{j+1}))}_{T_{j+2}} \\
&= T_1 \cup T_2 \cup \dots T_{j+1} \cup \Gamma(S_{j+1}) \\
&= \Gamma(S_0 \cup S_1 \cup \dots S_j) \cup \Gamma(S_{j+1}) \qquad \text{Por H.I} \\
&= \Gamma(S_0 \cup S_1 \cup \dots S_j \cup S_{j+1})
\end{aligned}$$

Sea $S = S_0 \cup S_1 \cup \dots S_k$

$$\begin{aligned}
|\Gamma(S)| &= |\Gamma(S_0 \cup S_1 \cup \dots S_k)| \\
&= |T_1 \cup T_2 \cup \dots \underbrace{T_{k+1}}_{=\emptyset}| && \text{Por Obs 3} \\
&= |T_1 \cup T_2 \cup \dots T_k| \\
&= |T_1| + |T_2| + \dots |T_k| && \text{Por Obs 2} \\
&= |S_1| + |S_2| + \dots |S_k| && \text{Por Obs 1} \\
&= |S| - |S_0| && \text{Por Obs 2} \\
&< |S| && \text{Pues } S_0 \neq \emptyset
\end{aligned}$$

Absurdo!

El absurdo vino de suponer que G es bipartito con $|S| \leq |\Gamma(S)| \; \forall S \subseteq X$ pero que no tiene matching completo de X a Y.

Q.E.D.

3.5. Teorema del matrimonio

Teorema: Todo grafo bipartito regular tiene un matching perfecto.
Prueba: Sea $G = (X \cup Y, E)$ bipartito regular con $E \neq \emptyset$, tal que $\forall\, W \subseteq V(G)$, definimos:

$$\begin{aligned} E_W &= \{xy \in E(G) : x \in W \ o\ y \in W\} \\ &= \{\text{lados con un extremo en W}\} \end{aligned}$$

Supongamos que $W \subseteq X$ (de igual forma para $W \subseteq Y$). Además, como G es regular, $\exists\, \Delta = \delta > 0 : d(z) = \Delta \ \ \forall z$.

$$\begin{aligned} |E_w| &= |\{xy \in E : x \in W\}| \\ &= \sum_{x \in w} |\{y : xy \in E\}| \\ &= \sum_{x \in w} \underbrace{d(x)}_{\Delta} \\ &= \Delta * |w| \end{aligned}$$

Es decir, a cada w le corresponden Δ lados distintos. En particular:

$$\begin{aligned} |E_x| &= \Delta * |x| \\ |E_y| &= \Delta * |y| \end{aligned}$$

pero $E_x = E = E_y$ pues G es bipartito.

Por lo tanto:

$$\begin{aligned} |E_x| = |E_y| &\Rightarrow \Delta * |x| = \Delta * |y| \\ &\Rightarrow |x| = |y| \\ &\Rightarrow \text{ Todo matching completo es perfecto.} \end{aligned}$$

Basta ver que existe un matching completo de X a Y, es decir, que se cumple la condición de Hall.

Sea $S \subseteq X$:

$$\text{Sea } \underbrace{xy}_{\substack{x \in X \\ y \in Y}} \in E_s \Rightarrow \left\{ \begin{array}{l} x \in S \\ y \in \Gamma(x) \end{array} \right\} \Rightarrow y \in \Gamma(S) \Rightarrow xy \in E_{\Gamma(s)}$$

Es decir, hemos probado que:

$$\begin{aligned} E_s &\subseteq E_{\Gamma(S)} \\ |E_S| &\leq |E_{\Gamma(S)}| \\ \Delta\, |S| &\leq \Delta\, |\Gamma(S)| \\ |S| &\leq \Gamma(S) \end{aligned}$$

Q.E.D.

3.6. Todo grafo bipartito es Δ coloreable

Teorema: Si G es bipartito $\Rightarrow \chi'(G) = \Delta$
Prueba:

Lema Interno: Todo grafo bipartito es subgrafo de un grafo bipartito regular con el mismo Δ, es decir:

$$\text{G bipartito} \Rightarrow \exists \text{ H bipartito regular con G} \subseteq \text{H y } \Delta(G) = \Delta(H)$$

Prueba: Sean:

- $G = (V = X \cup Y, E)$ grafo bipartito
- $G^\star = (V^\star, E^\star)$ una copia de G
- $E^\dagger = \{vv^\star : d_G(v) < \Delta(G)\}$
- $\overline{G} = (\overline{V}, \overline{E})$ con:
  - $\overline{V} = V \cup V^\star$
  - $\overline{E} = E \cup E^\star \cup E^\dagger$

Propiedades de $\overline{G}$:

1. $\overline{G}$ es bipartito, sus partes son:
   - $X \cup Y^\star$
   - $X^\star \cup Y$

   No existen lados entre $x \leftrightarrow x,\ y^\star \leftrightarrow y^\star,\ x \leftrightarrow y^\star, x^\star \leftrightarrow x^\star,\ y \leftrightarrow y,\ x^\star \leftrightarrow y^\star$.

2. Sea v $\in$ V tal que $d_G(v) = \Delta = \Delta(G) \Rightarrow v$ no es parte de ningún lado de $E^\dagger$.

   $$\therefore d_{\overline{G}}(v) = d_{\overline{G}}(v^\star) = \Delta$$

   Sea v $\in$ V tal que $d_G(v) < \Delta \Rightarrow v$ forma parte de un lado de $E^\dagger$.

   $$\therefore d_{\overline{G}}(v) = d_{\overline{G}}(v^{\star)} = d_G(v) + 1$$

   <u>Conclusión:</u>

   $$\begin{aligned} \Delta(\overline{G}) &= \Delta(G) \\ \delta(\overline{G}) &= \delta(G) + 1 \end{aligned}$$

   Repitiendo este proceso, *i.e* $G \rightarrow \overline{G} \rightarrow \overline{\overline{G}} \rightarrow \overline{\overline{\overline{G}}} \ldots$ eventualmente llegaremos a un $G^\blacktriangle$ tal que $\delta(G^\blacktriangle) = \Delta\ \therefore$ regular.

Fin del Lema Interno

- Sea H bipartito regular con $G \subseteq H$ y $\Delta(G) = \Delta(H)$. Como H es bipartito regular $\Rightarrow \exists$ un matching perfecto en H, llamado M. Coloreamos todos los lados de M con **`Color 1`**.
- Sea $H_1 = H - \{\text{lados del matching M}\}$. Como M es matching perfecto, H_1 sigue siendo regular y $\delta_{H_1}(x) = \delta_H(x) - 1 = \Delta - 1$. Como H_1 es bipartito regular $\Rightarrow \exists$ un matching perfecto en H_1, llamado M_1. Coloreamos todos los lados de M_1 con **`Color 2`**.
- Sea $H_2 = H_1 - \{\text{los lados del matching } M_1\}$. Luego $\delta_{H_2}(x) = \delta_{H_1}(x) - 1 = \Delta - 2$.

  Siguiendo así $H_{\Delta-1}$ seguirá siendo regular con $\delta_{H_{\Delta-1}}(x) = 1$. En total obtuvimos Δ matchings y Δ colores. $\therefore \chi'(H) = \Delta$

$$\Rightarrow \chi'(G) = \Delta \text{ pues } G \subseteq H.$$

Q.E.D.

3.7. Cota de Hamming

Teorema: Sea C un código de longitud n y sea $t = \lfloor \frac{\delta-1}{2} \rfloor$ la cantidad de errores que corrigue, entonces:

$$|C| \;\leq\; \frac{2^n}{1 + n + \binom{n}{2} + \ldots \binom{n}{t}}$$

Prueba: Sea $\boxed{A = \cup_{x \in C} B_t(x)}$

Como ya dijimos C corrigue t errores $\Rightarrow B_t(x) \cap B_t(y) \neq \emptyset \; \forall x, y \in C : x \neq y$.

$$\therefore \text{La unión en A es disjunta y } \boxed{|A| = \textstyle\sum_{x \in C} |B_t(x)|} \text{ (1)}$$

Sea $S_r(x) = \{y \in Z_2^n : d(x,y) = r\}$

$\therefore B_t(x) = \cup_{r=0}^{t} S_r(x)$ y la unión es disjunta $\Rightarrow \boxed{|B_t(x)| = \sum_{r=0}^{t} |S_r(x)|}$ (2)

¿Cuánto vale $S_r(x)$?

$$\begin{aligned} y \in S_r(x) &\Leftrightarrow d(x,y) = r \\ &\Leftrightarrow |x \oplus y| = r \text{ como } y = x \oplus \underbrace{(x \oplus y)}_{\epsilon} \\ &\Leftrightarrow \exists \underbrace{\epsilon : |\epsilon| = r}_{\in S_r(0)} : y = x \oplus \epsilon \end{aligned}$$

$\therefore \; S_r(x) = x \oplus S_r(0) \Rightarrow |S_r(x)| = |S_r(0)|$

Luego

$$\begin{aligned} |S_r(x)| &= |S_r(0)| \\ &= \# \text{ vectores de longitud n con r unos} \\ |S_r(x)| &= \binom{n}{r} \qquad (3) \end{aligned}$$

Por lo tanto

$$\begin{aligned} |A| &= \sum_{x \in C} |B_t(x)| && \text{Por (1)} \\ &= \sum_{x \in C} \sum_{r=0}^{t} |S_r(x)| && \text{Por (2)} \\ &= \sum_{x \in C} \sum_{r=0}^{t} \binom{n}{r} && \text{Por (3)} \\ &= |C| * \sum_{r=0}^{t} \binom{n}{r} \end{aligned}$$

Como $A \in Z_2^n \Rightarrow |A| \leq 2^n$, entonces:

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

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

Q.E.D.

3.8. Teorema de la matriz de chequeo de códigos lineales

Teorema: Sea C un código lineal con matriz de chequeo H, entonces:

a) $\delta = \delta(C) = \text{mín}\,\{|x| : x \in C, x \neq 0\}$ (#columnas LD de H).

b) Si H no tiene columnas 0, ni columnas repetidas, entonces $\delta \geq 3$ y corrigue al menos un error.

Prueba: Sean:

- $w =$ mín #columnas LD de H.
- $e_i = 0\dots0 \underbrace{1}_{i} 0\dots0$

a)

$\boxed{w \leq \delta}$

Sea $v \in C, v \neq 0$ y $|v| = \delta \Rightarrow v$ tiene δ unos. Es decir; $\exists j_1, j_2 \dots j_\delta$ tal que:

$$v = e_{j_1} + e_{j_2} + \dots e_{j_\delta}$$

Como $v \in C \Rightarrow v =$ NU(H) entonces $Hv^t = 0$.

$$\begin{aligned} 0 = Hv^t &= H(e^t_{j_1} + e^t_{j_2} + \dots e^t_{j_\delta}) \\ &= He^t_{j_1} + He^t_{j_2} + \dots He^t_{j_\delta} \\ &= H^{j_1} + H^{j_2} + \dots H^{j_\delta} \end{aligned}$$

$\therefore \{H^{j_1}, H^{j_2}, \dots H^{j_\delta}\}$ es LD $\Rightarrow w \leq \delta$.

$\boxed{w \geq \delta}$

Sea ahora $\{H^{i_1}, H^{i_2}, \dots H^{i_w}\}$ un conjunto LD, entonces $\exists$ constantes $c_w \in Z_2^n$ no todas nulas, tales que:

$$c_1H^{i_1} + c_2H^{i_2} + \dots c_wH^{i_w} = 0$$

Sea $v = c_1e_{i_1} + c_2e_{i_2} + \dots c_we_{i_w}$, $v \neq 0$ ya que dijimos que no todos los c_w eran nulos.

Luego:

$$\begin{aligned} Hv^t &= H(c_1e^t_{i_1} + c_2e^t_{i_2} + \dots c_we^t_{i_w}) \\ &= Hc_1e^t_{i_1} + Hc_2e^t_{i_2} + \dots Hc_we^t_{i_w} \\ &= H^{i_1} + H^{i_2} + \dots H^{i_w} \\ &= 0 \end{aligned}$$

$\Rightarrow v \in C$ y $|v| = w$. Como δ es la menor distancia, tenemos que $\delta \leq |v| = w$.

Por lo tanto vale $\boxed{w \leq \delta}$ y $\boxed{w \leq \delta} \Rightarrow \boxed{w = \delta}$.

b)

- Si H no tiene columnas ceros $\Rightarrow w \geq 2$.
- Si H no tiene columnas repetidas $\Rightarrow \nexists i, j : i \neq j$ y $H^i = H^j$. Como no puede pasar que $Hi = Hj$ tampoco $Hi + Hj = 0$, es decir, $Hi + Hj \neq 0$ $\forall i, j : i \neq j$ por lo que $\nexists$ conjuntos de columnas LD $\Rightarrow w \geq 3$.

  Luego $\delta \geq 3 \Rightarrow t = \lfloor \frac{\delta - 1}{2} \rfloor \geq 1$, es decir, C corrigue al menos un error.

Q.E.D.

3.9. Teorema del polinomio generador de códigos cíclicos

Teorema: Sea C un código cíclico de longitud n y dimensión k y sea $g(x)$ su polinomio generador, entonces:

a) C está formado por los múltiplos de $g(x)$ de grado menor a n.

b) El grado de $g(x)$ es $n-k$.

c) $g(x)$ divide a $1+x^n$

Prueba:

a) Sea v(x) $\in$ C, dividamos v(x) por g(x). Entonces $\exists$ q(x) y r(x) con $gr(r(x)) < gr(g(x))$ tal que:

$$\begin{aligned} v(x) &= q(x)g(x) + r(x) \\ &\Rightarrow \boxed{q(x)g(x) = v(x) + r(x)}\ (1) \end{aligned}$$

.

Como v(x) $\in$ C $\Rightarrow gr(v(x)) < n$ y $gr(r(x)) < gr(g(x)) \underbrace{<}_{g(x)\in C} n$, concluimos que:

$$\boxed{gr(v(x)+r(x)) < n}\ (2)$$

Luego de (1) y (2) deducimos: $\boxed{gr(q(x)\ g(x)) < n}$

Recordemos que si $p(x) \in$ C y $gr(p(x)) < n$ entonces: $\boxed{p(x)\ mod(1+x^n) = p(x)}$

Por lo tanto:

$$q(x)\ g(x)\ mod\ (1+x^n) = q(x)\ g(x)$$

$$\text{i.e} \boxed{q(x) \odot g(x) = q(x)\ g(x)}\ \text{(A)}$$

Además como $g(x) \in$ C $\Rightarrow \boxed{q(x) \odot g(x) \in \text{C}}$ (B)

De (A) y (B) resuta que $q(x)\ g(x) \in$ C $\Rightarrow v(x) + r(x) \in C$. Además dijimos que v(x) $\in$ C y como C es lineal $\Rightarrow \boxed{\text{r(x)} \in C}$ (†)

Llamemos $\boxed{gr(r(x)) < gr(g(x))}$ ($\star$)

De (†) y ($\star$) deducimos $r(x) = 0 \Rightarrow v(x) = q(x)\ g(x)$.

b) Recordemos que si v(x) $\in$ C $\Rightarrow \exists\ q(x) : v(x) = q(x)g(x)$. Entonces para que $gr(q(x)g(x)) < n$ debe darse que $gr(q(x)) + gr(g(x)) < n$, es decir, $gr(q(x)) < n - gr(g(x))$.

Sea:

$$C = \{v(x) : \exists\ q(x),\ gr(q(x)) < n - gr(g(x)),\ v(x) = q(x)g(x)\}$$

Entonces existe una biyección entre C y el conjunto: $\{q(x) : gr(q(x)) < n - gr(g(x))\}$.

$$\begin{aligned} \therefore\ |C| &= |\{q(x) : gr(q(x)) < n - gr(g(x))\}| \\ 2^k &= 2^{n-gr(g(x))} \\ k &= n - gr(g(x)) \\ gr(g(x)) &= n-k \end{aligned}$$

c) Dividamos $1 + x^n$ por g(x).

$$\begin{array}{rcl} 1 + x^n & = & q(x)g(x) + r(x) \qquad \text{Con } gr(r(x)) < gr(g(x)) \\ \therefore r(x) & = & q(x)g(x) + (1 + x^n) \end{array}$$

Tomando $mod(1 + x^n)$:

$$\begin{array}{rcl} r(x)\ mod\ (1 + x^n) & = & (q(x)g(x) + (1 + x^n))\ mod\ (1 + x^n) \\ \therefore r(x) & = & q(x) \odot g(x) \in C \end{array}$$

$$\left.\begin{array}{l} r(x) \in C \\ gr(r(x)) < gr(g(x)) \end{array}\right\} \Rightarrow r(x) = 0$$

$$\therefore\ g(x) \mid 1 + x^n$$

Q.E.D.

Bibliografía

[1] Maximiliano Illbele, «Resumen de Discreta II, 16 de agosto de 2012», *FaMAF, UNC.*

[2] Lucia Pappaterra, «Resumen de Discreta II, 2014», *FaMAF, UNC.*

[3] Marcos Modenesi, «Resumen de Discreta II, 2016», *FaMAF, UNC.*

[4] Agustín Curto, «Carpeta de Clase, 2016», *FaMAF, UNC.*