Resumen de teoremas para el final de Matemática Discreta II

Agustin Curto, agucurto95@gmail.com

2016

Índice general

Capítulo 1

Parte A

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:

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

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: $s = z_0,\ z_1,\ \ldots\ z_i,\ z_{i+1} = z$.
   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 será $\leq$ A.

- Caso 2: 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 $s,\ \ldots\ z,\ x$.
  Como $\langle E - K \rangle$ funciona con BFS, ese camino usado para construir f_k debe ser de longitud mínima. Es decir:

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

Conclusión: En cualquiera de los dos casos tenemos:

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

Ahora bien:

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

Por lo tanto:

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

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 $\mathcal{O}(nm)$.

Sea:

- 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**

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

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

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:

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

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 saturarse 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" este bloqueado (porque BB(y) solo se ejecuta si "y" está bloqueado), pero si "y" está bloqueado "x" no puede mandarle flujo nunca más.

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

  Conclusión 1: Los lados se saturan solo una vez.

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

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

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

  Conclusión 2: 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.

1.5. La distancia entre NA sucesivos aumenta

Teorema: Sea A un NA (network auxiliar) y sea A^* el siguiente NA. Sean $d(x)$ y $d^*(x)$ las distancias de s a x en A y A^* respectivamente, entonces: $d(t) < d^*(t)$.

Prueba: Como A y A^* se construyen con BFS sabemos que $d(t) \leq d^*(t)$ pero queremos ver el $<$.

Sea:

$$s = x_0, x_1, \ldots x_r = t$$

un camino dirigido en A^*.

Ese camino $\boxed{\text{NO EXISTE}}$ en A ya que para pasar de A a A^* debemos bloquear todos los caminos dirigidos de A. Por lo tanto si ese camino estuviese en A, Dinic lo habría bloqueado y no estaría en A^*.

¿Cuáles son las razones posibles para que no esté en A?

1. Puede faltar un vértice, es decir $\exists i : x_i \notin V(A)$ entonces:

$$\begin{aligned} d(t) &\leq d(x_i) \quad \text{Por def de } d_k(x),\ \not\exists \infty \\ &\leq d^*(x_i) \quad \text{Por } \langle E - K \rangle \\ &< d^*(t) \quad \text{Porque } x_i \text{ esta antes que t} \end{aligned}$$

$$\boxed{\therefore\ d(t) < d^*(t)}$$

2. Están todos los vértices pero falta una arista, es decir $\exists i : \overrightarrow{x_i x_{i+1}} \notin E(A)$.

   a) $\overrightarrow{x_i x_{i+1}}$ no está porque corresponde a un lado vacío o saturado en NA, es decir $\overrightarrow{x_i x_{i+1}}$ no está en el residual que dá origen a A pero si está en el residual que dá origen a A^*. Para que esto pase se tiene que haber usado el lado $\overrightarrow{x_{i+1} x_i}$ en A. Luego podemos concluir, por la prueba de $\langle E - K \rangle$ que:

$$d^*(t) \geq d(t) + 2 > d(t)$$

$$\boxed{\therefore\ d(t) < d^*(t)}$$

   b) $\overrightarrow{x_i x_{i+1}}$ si está en el residual pero: $\boxed{d(x_{i+1}) \neq d(x_i) + 1}$ (1)

   Pero como $\overrightarrow{x_i x_{i+1}}$ está en el residual entonces: $\boxed{d(x_{i+1}) \leq d(x_i) + 1}$ (2)

   De (1) y (2) tenemos que: $\boxed{d(x_{i+1}) < d(x_i) + 1}$ $\circledast$

   Entonces:

$$\begin{aligned} d(t) &= d(x_{i+1}) + b(x_{i+1}) && \text{Por } \langle E - K \rangle \\ &\leq d(x_{i+1}) + b^*(x_{i+1}) && \text{Por } \langle E - K \rangle \\ &< d(x_i) + 1 + b^*(x_{i+1}) && \text{Por } \circledast \\ &\leq d^*(x_i) + 1 + b^*(x_{i+1}) && \text{Por } \langle E - K \rangle \\ &= d^*(x_{i+1}) + b^*(x_{i+1}) && \text{Por } (\dagger) \\ &= d^*(t) \end{aligned}$$

$$\boxed{\therefore\ d(t) < d^*(t)}$$

   ($\dagger$): Ya que $s,\ x_1,\ \ldots\ x_r$ es un camino en A*.

Q.E.D.

Capítulo 2

Parte B

2.1. 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 $\mathcal{O}(m)$ (recorrer todas las aristas).
  - (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 $C(y) = C(z) \Rightarrow Nivel_{BFS(x)}(y) = Nivel_{BFS(x)}(z)$

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 $t = Nivel_{BFS(x)}(z)$ $(s \equiv t)$.
- 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: $w \dots \overbrace{yz}^{Por\ Hipotesis} \dots w$

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

$\therefore$ Es un ciclo impar

Q.E.D.

2.2. Teorema Max-Flow Min-Cut

Teorema:

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).

Prueba: Denotemos ($\star$), a la demostración de:

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

a) f es flujo y S es corte $\Rightarrow$ V(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(f) = Cap(S) $\Rightarrow$ 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 $x\}$

$\boxed{¿t \in S?}$

Si t estuviera en S: 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 $s \dots x$.

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

En particular $s \dots x \dots y$ 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 $s \dots y$.

En particular $s \dots y \dots x$ 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.

2.3. La complejidad del HÚNGARO

Teorema: La complejidad del algoritmo Húngaro es $\mathcal{O}(n^4)$.
Prueba:

1. La complejidad del matching inicial es $\mathcal{O}(n^2)$, ya que:

   Restar mínimo de cada fila:

   $$(\underbrace{\mathcal{O}(n)}_{calcular\ min} + \underbrace{\mathcal{O}(n)}_{restar\ min}) * \underbrace{n}_{\#filas} = \mathcal{O}(n^2)$$

   Idem para las columnas.

2. Llamemos **extender** el matching, a incrementar su número de filas en 1, es decir, agregar una fila más al matching.

   $$\#\text{ extensiones de matching} = \mathcal{O}(n)$$

   Resta ver la complejidad de cada **extender**.

3. En cada extensión vamos a ir revisando filas y columnas, donde escanear una fila es $\mathcal{O}(n)$ y se realizan n escaneos, por lo que sería $\mathcal{O}(n^2)$ sin considerar que se debe realizar un cambio de matriz.

   Hacer un **cambio de matriz** es $\mathcal{O}(n^2)$, ya que:

   - Buscar $m = \text{mín}\, S$ x $\overline{\Gamma(S)} \rightarrow \mathcal{O}(n^2)$
   - Restar m de $S \rightarrow \mathcal{O}(n^2)$
   - Sumar m a $\Gamma(S) \rightarrow \mathcal{O}(n^2)$

   Luego la implementación NAIVE lanzaría nuevamente el algoritmo desde cero. La forma correcta es continuar con el matching que teniamos, ya que el mismo no se pierde.

   Si lo hacemos así, ¿Cuántos **cambios de matriz** habrá antes de extender un matching nuevamente?

   Lema Interno: Luego de un **cambio de matriz**, se extiende el matching o bien se aumenta S.

   Prueba:

   $$\begin{array}{c} \left(\begin{array}{c|c} \text{A} & \text{A} \\ \hline \text{B} & \text{C} \end{array}\right) \begin{array}{c} \overline{\text{S}} \\ \text{S} \end{array} \\ \Gamma(S)\ \overline{\Gamma(S)} \end{array}$$

   Referencias:

   - A: puede haber ceros.
   - B: ceros del matching.
   - C: no hay ceros, no hay matching.

   Al restar $m = \text{mín}\, S$ x $\overline{\Gamma(S)}$ de las filas de S, habrá un nuevo cero en alguna fila i ($\in S$) y columna j ($\in \overline{\Gamma(S)}$) entonces la columna se etiquetará con i y se revisará. Tenemos dos resultados posible:

a) j está libre (i.e no forma parte del matching) $\Rightarrow$ extendemos el matching.

b) j forma parte de matching $\Rightarrow \exists$ fila k matcheada con j. En este caso, la fila k se etiquetará con j, por lo que el "nuevo" $S \geq S \cup \{k\}$.

Entonces se termina con una extensión o se produce un nuevo S de cardinalidad, al menos $|S| + 1$.

Fin Lema Interno

Luego como $|S|$ solo puede crecer $\mathcal{O}(n)$ veces, tenemos que hay a lo sumo n **cambios de matriz** antes de extender el matching. Entonces:

$$\text{Complejidad(1 extensión)} = \underbrace{\mathcal{O}(n)}_{\#CM} * \underbrace{\mathcal{O}(n^2)}_{Complejidad(CM)} + \underbrace{\mathcal{O}(n^2)}_{Busqueda\ n\ filas\ x\ n\ columnas}$$

$$\begin{aligned} \therefore \text{Complejidad(Hungaro)} &= \text{Compl(Matching inicial)} + (\#\text{extensiones} * \text{Compl(1 extensión)}) \\ &= \mathcal{O}(n^2) + (\mathcal{O}(n) * \mathcal{O}(n^3)) \\ &= \mathcal{O}(n^4) \end{aligned}$$

Q.E.D.

2.4. Teorema de Hall

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

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

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. $|S_j| = |T_j|$, pues S_j son las filas matcheadas con T_j.
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 $S = S_0 \cup S_1 \cup \dots S_k$

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

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.

2.5. 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.

2.6. 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
- $E^\dagger = \{vv^\star : d_G < \Delta(G)\}$
- $\overline{G} = (\overline{V}, \overline{E})$ 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:
   - $X \cup Y^\star$
   - $X^\dagger \cup Y$

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

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

   Sea v $\in$ V tal que $d_G(v) < \Delta \Rightarrow$ forma parte de un nuevo lado.
   $\therefore d_{\overline{G}}(v) = d_{\overline{G}}(v^{\star)} = d_G(v) + 1$

   Conclusión:

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

   Repitiendo este proceso, *i.e* $G \to \overline{G} \to \overline{\overline{G}} \to \overline{\overline{\overline{G}}} \ldots$ 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. Coloreemos todos los lados de M con `Color 1`.

- Sea $H_1 = H-$ los 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. Coloreemos todos los lados de M_1 con `Color 2`.

- Sea $H_2 = H_1-$ 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.

2.7. 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 $\Rightarrow B_t(x) \cap B_t(y) \neq \emptyset \ \forall x, y \in C : x \neq y$.

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

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 $S_r(x)$?

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

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

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.

2.8. 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 corrige al menos un error.

Prueba: Sean:

- $w = \text{mín}$ #columnas LD de H.
- $e_i = 000\ldots\underbrace{1}_{i}0\ldots0$

a)

$\boxed{w \leq \delta}$ Sea $v \in C, v \neq 0$ y $|v| = \delta \Rightarrow v$ tiene δ unos.

Es decir; $\exists j_1, j_2 \ldots j_\delta$ tal que:

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

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

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

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

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

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

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

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

$\Rightarrow v \in C$ 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 corrige al menos un error.

Q.E.D.

2.9. Teorema del polinomio generador de códigos cíclicos

Teorema: Sea C un código cíclico de dimensión k y longitud n 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 $gr(r(x)) < gr(g(x)) \underbrace{<}_{g(x)\,\in\, C} n$ con lo que 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).

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

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

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

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

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

Q.E.D.

Capítulo 3

Parte C

3.1. 4-COLOR $\leq_p$ SAT

Teorema: 4-COLOR $\leq_p$ SAT.
Prueba: Sea G un grafo con vértices $v_1, v_2, \ldots v_n$, queremos construir (en tiempo polinomial) una expresión booleana B en CNF (*forma conjuntiva normal*), tal que:

$$\chi(G) \leq 4 \Leftrightarrow \text{ B es satisfacible}$$

Sea n = #vértices de G, y sean las variables $x_{i,j}$ con $i = 1, 2, \ldots n$ y $j = 1, 2, 3, 4$

Sean:

$$\begin{aligned} A_i &= x_{i,1} \vee x_{i,2} \vee x_{i,3} \vee x_{i,4} \\ A &= A_1 \wedge A_2 \wedge \ldots A_n \end{aligned}$$

Defino:

$$\begin{aligned} Q_{i,j,k,l} &= \overline{x_{i,j}} \vee \overline{x_{k,l}} \qquad i,k = 1 \ldots n \qquad j,l = 1,2,3,4 && (1) \\ D_i &= Q_{i,i,1,2} \wedge Q_{i,i,1,3} \wedge Q_{i,i,1,4} \wedge Q_{i,i,2,3} \wedge Q_{i,i,2,4} \wedge Q_{i,i,3,4} && (2) \\ D &= D_1 \wedge D_2 \ldots \wedge D_n && (3) \\ F_{i,k} &= Q_{i,k,1,1} \wedge Q_{i,k,2,2} \wedge Q_{i,k,3,3} \wedge Q_{i,k,4,4} && (4) \\ F &= \wedge_{\substack{i,k \\ v_i v_k \in E(G)}} F_{i,k} && (5) \end{aligned}$$

$\boxed{\Rightarrow}$

Sea C un coloreo propio de G, con a lo sumo 4 colores. Debemos dar un asignamiento de valores a las variables $x_{i,j}$ que satisfagan B, es decir, un elemento $\overrightarrow{b} \in \mathbb{Z}_2^{4n}$ tal que $B(\overrightarrow{b}) = 1$.

Definimos

$$b_{i,j} \begin{cases} 1 \text{ si } C(v_i) = j \\ 0 \ c.c \end{cases}$$

Debemos probar $B(\overrightarrow{b}) = 1$, es decir, (1) $A(\overrightarrow{b}) = 1$, (2) $D(\overrightarrow{b}) = 1$ y (3) $F(\overrightarrow{b}) = 1$.

(1) $\boxed{A(\overrightarrow{b}) = 1}$
Debemos ver que $A_i(\overrightarrow{b}) = 1 \ \forall i$, es decir, hay que probar:

$$b_{i,1} \vee b_{i,2} \vee b_{i,3} \vee b_{i,4} = 1$$

pero

$$C(v_i) \in \{1,2,3,4\} \Rightarrow \exists j \in \{1,2,3,4\} \text{ tal que } C(v_i) = j \Rightarrow b_{i,j} = 1$$

(2) $\boxed{D(\overrightarrow{b}) = 1}$

Debemos ver que $D_i(\overrightarrow{b}) = 1 \; \forall i$, es decir,

$$Q_{i,j,j,l}(\overrightarrow{b}) = 1 \; \forall i, j, l, j < l$$

es decir:

$$\overline{b_{i,j}} \vee \overline{b_{i,l}} = 1 \; \forall i, j, l, j < l$$

por el absurdo, supongamos que no:

$$\begin{aligned}
&\Rightarrow \quad \exists i, j, l, j < l \; : \; \overline{b_{i,j}} \vee \overline{b_{i,l}} = 0 \\
&\Rightarrow \quad \overline{b_{i,j}} = \overline{b_{i,l}} = 0 \\
&\Rightarrow \quad b_{i,j} = b_{i,l} = 1 \\
&\Rightarrow \quad C(v_i) = j, \; C(v_i) = l, \; \textit{pero } j \neq l \text{ Absurdo!}
\end{aligned}$$

Luego, el absurdo vino de suponer que $\overline{b_{i,j}} \vee \overline{b_{i,l}} = 0$

$\therefore \overline{b_{i,j}} \vee \overline{b_{i,l}} = 1$

(3) $\boxed{F(\overrightarrow{b}) = 1}$

Por el absurdo, supongamos que no:

$$\begin{aligned}
&\Rightarrow \quad \exists i, k \; : \; v_i v_k \in E(G) \text{ tal que } F_{i,k}(\overrightarrow{b}) = 0 \\
&\Rightarrow \quad \exists j \; \in \{1,2,3,4\} : Q_{i,k,j,j}(\overrightarrow{b}) = 0 \\
&\Rightarrow \quad \overline{b_{i,j}} \vee \overline{b_{k,j}} = 0 \\
&\Rightarrow \quad \overline{b_{i,j}} = \overline{b_{k,j}} = 0 \\
&\Rightarrow \quad b_{i,j} = b_{k,j} = 1 \\
&\Rightarrow \quad C(v_i) = 1 \; y \; C(v_k) = 1 \\
&\Rightarrow \quad C(v_i) = C(v_k) \; (= j) \text{ Absurdo!}
\end{aligned}$$

Pues, el coloreo C es propio y $v_i v_k \in E(G)$. El absurdo vino de suponer que $F(\overrightarrow{b}) = 0$

$\therefore F(\overrightarrow{b}) = 1$

$\boxed{\Leftarrow}$

Ahora sabemos que existe un $\overrightarrow{b}$ con $B(\overrightarrow{b}) = 1$ y debemos construir un coloreo propio, con al menos 4 colores.

$$B(\overrightarrow{b}) = 1 \Rightarrow \begin{cases} A(\overrightarrow{b}) = 1 \Rightarrow A_i(\overrightarrow{b}) = 1 \; \forall i \Rightarrow (1) \; \forall i \; \exists j : b_{i,j} = 1 \\ D(\overrightarrow{b}) = 1 \Rightarrow D_i(\overrightarrow{b}) = 1 \; \forall i \Rightarrow (2) \; \forall i \; \nexists j \neq l : b_{i,j} = 1 = b_{i,l} \\ F(\overrightarrow{b}) = 1 \Rightarrow \forall i, j \in E(G), C(v_i) \neq C(v_j) \Rightarrow \text{C es propio} \end{cases}$$

De (1) y (2) $\Rightarrow \boxed{\forall i \; \exists! j : b_{i,j} = 1}$

Luego, definimos:

$$C(v_i) = \text{unico j con } b_{i,j} = 1$$

Q.E.D.

3.2. 3-SAT es NP-Completo

Teorema: 3-SAT es NP-Completo.
Prueba: Probaremos que SAT $\leq_p$ 3-SAT, pues 3-SAT es NP Completo.

Sea

$$B = D_1 \wedge D_2 \ldots \wedge D_n \text{ con variables } x_1, \ldots x_n \tag{6}$$

$$D_i = L_{i,1} \vee L_{i,2} \ldots \vee L_{i,k_i} \tag{7}$$

$$l_{i,j} = \text{literales} \tag{8}$$

Para cada D_i construiremos E_i, que serán conjunciones de disyunciones de 3 literales, con variables extras, y luego tomaremos:

$$B' = E_1 \wedge E_2 \ldots \wedge E_n$$

Construcción:

- Si D tiene 3 literales, es decir, $K_i = 3 \Rightarrow E_i = D_i$
- Si D tiene 2 literales, es decir, $K_i = 2 \Rightarrow$ tomo una variable extra $y_{i,1}$ y defino:

$$E_i = (l_{i,1} \vee l_{i,2} \vee y_{i,1}) \wedge (l_{i,1} \vee l_{i,2} \vee \overline{y_{i,1}})$$

- Si D tiene 1 literales, es decir, $K_i = 1 \Rightarrow$ tomo dos variables extras $y_{i,1}$ y $y_{i,2}$ y defino:

$$E_i = (l_{i,1} \vee y_{i,1} \vee y_{i,2}) \wedge (l_{i,1} \vee \overline{y_{i,1}} \vee y_{i,2}) \wedge (l_{i,1} \vee y_{i,1} \vee \overline{y_{i,2}}) \wedge (l_{i,1} \vee \overline{y_{i,1}} \vee \overline{y_{i,2}})$$

- Si D tiene más de 3 literales, es decir, $K_i > 4 \Rightarrow$ agrego k_{i-3} variables extras y defino:

$$E_i = (l_{i,1} \vee l_{i,2} \vee y_{i,1}) \ \wedge \ (l_{i,3} \vee \overline{y_{i,1}} \vee y_{i,2}) \ \wedge \ (l_{i,4} \vee \overline{y_{i,2}} \vee y_{i,3}) \wedge$$

$$\ldots (l_{i,k_{i-2}} \vee \overline{y_{i,k_{i-4}}} \vee y_{i,k_{i-3}}) \wedge (l_{i,k_{i-1}} \vee l_{i,k_i} \vee \overline{i, k_{i-3}})$$

Queremos ver:

$$B(\overrightarrow{b}) = 1 \Leftrightarrow \exists \overrightarrow{a} : B'(\overrightarrow{b}, \overrightarrow{a}) = 1$$

$\boxed{\Leftarrow}$
Por el absurdo. Supongamos que no, entonces $B(\overrightarrow{b}, \overrightarrow{a}) = 1$ pero $B(\overrightarrow{b}) = 0$.
Si $B(\overrightarrow{b}) = 0 \Rightarrow \exists i : D_i(\overrightarrow{b}) = 0$ pero $E_i(\overrightarrow{b}, \overrightarrow{a}) = 1$.

Casos:

- si $k_i = 3$ Absurdo! pues $D_i = E_i$
- si $k_i = 2$

$$\begin{aligned} 1 &= E_i(\overrightarrow{b}, \overrightarrow{a}) \\ &= (\underbrace{l_{i,1}(\overrightarrow{b}) \vee l_{i,2}(\overrightarrow{b})}_{=0} \vee a_{i,1}) \wedge (\underbrace{l_{i,1}(\overrightarrow{b}) \vee l_{i,2}(\overrightarrow{b})}_{=0} \vee \overline{a_{i,1}}) \\ &= a_{i,1} \wedge \overline{a_{i,1}} \\ &= 0 \text{ Absurdo!} \end{aligned}$$

- si $k_i = 1$

$$\begin{aligned} 1 &= E_i(\overrightarrow{b}, \overrightarrow{a}) \\ &= (a_{i,1} \vee a_{i,2}) \wedge (\overline{a_{i,1}} \vee a_{i,2}) \wedge (a_{i,1} \vee \overline{a_{i,2}}) \wedge (\overline{a_{i,1}} \vee \overline{a_{i,2}}) \\ &= 0 \text{ Absurdo!} \end{aligned}$$

- si $k_i > 4$

  $D_i(\overrightarrow{b}) = 0 \Rightarrow l_{i,j}(\bar{b}) = 0 \; \forall j.$

$$\begin{aligned} 1 &= E_i(\overrightarrow{b}, \overrightarrow{a}) \\ &= a_{i,j} \wedge (\overline{a_{i,1}} \vee a_{i,2}) \wedge (\overline{a_{i,2}} \vee a_{i,3}) \wedge \ldots (\overline{a_{i,k_{i-4}}} \vee a_{i,k_{i-3}}) \wedge \overline{a_{i,k_{i-3}}} \\ &= a_{i,1} \wedge (a_{i,1} \Rightarrow a_{i,2}) \wedge (a_{i,2} \Rightarrow a_{i,3}) \wedge \ldots (a_{i,k_{i-4}} \Rightarrow a_{i,k_{i-3}}) \wedge \overline{a_{i,k_{i-3}}} \text{ Absurdo!} \end{aligned}$$

  Pues tendría $a_{i,1} = a_{i,2} = \ldots a_{i,k_{i-3}} = 1$ pero $\overline{a_{i,k_{i-3}}} = 1$.

$\boxed{\Rightarrow}$

Casos:

- Si $k_i = 3 \Rightarrow E_i(\overrightarrow{b}, \overrightarrow{a}) = D_i(\overrightarrow{b}) = 1$
- Si $k_i = 1$ ó $k_i = 2$ defino $a_{i,j} = 0,$ luego $E_i(\overrightarrow{b}, 0) = 1$
- Si $k_i > 4 \Rightarrow D_i(\overrightarrow{b}) = 1 \Rightarrow \exists j : l_{i,j}(\overrightarrow{b}) = 1$

  Definimos

$$a_{i,1} = a_{i,2} = \ldots a_{i,j-2} = 1$$
$$a_{i,j-1} = a_{i,j} = \ldots a_{i,k-3} = 0$$

  Luego

$$\begin{aligned} E_i(\overrightarrow{b}, \overrightarrow{a}) \quad &= \quad (l_{i,1}(\overrightarrow{b}) \vee l_{i,2}(\overrightarrow{b}) \vee a_{i,1}) \\ &\wedge \quad (l_{i,3}(\overrightarrow{b}) \vee \overline{a_{i,1}} \vee a_{i,2}) \\ &\vdots \\ &\wedge \quad (l_{i,j-1}(\overrightarrow{b}) \vee \overline{a_{i,j-3}} \vee a_{i,j-2}) \\ &\wedge \quad (l_{i,j}(\overrightarrow{b}) \vee \overline{a_{i,j-2}} \vee a_{i,j-1}) \\ &\wedge \quad (l_{i,j+1}(\overrightarrow{b}) \vee \overline{a_{i,j-1}} \vee a_{i,j}) \\ &\vdots \\ &\wedge \quad (l_{i,k_{i-1}}(\overrightarrow{b}) \vee l_{i,k_i}(\overrightarrow{b}) \vee \overline{a_{i,k_{i-3}}}) \\ &\therefore \quad E_i = 1 \end{aligned}$$

Q.E.D.

3.3. 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 disjunción, debemos crear un grafo G, tal que:

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

Sea:

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

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:

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

Donde:

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

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

$\boxed{\Leftarrow}$

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:

$$b_k = \begin{cases} 1 \text{ si } C(v_k) = C(s) \\ 0 \text{ si } C(v_k) \neq C(s) \end{cases}$$

Para probar $B(\overrightarrow{b}) = 1$ debemos probar que $D_i(\overrightarrow{b}) = 1 \forall i$.

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.

Por lo tanto: $\exists j : C(b_{i,j}) = C(s)$

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:

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

- Caso (1):

$$\begin{aligned} l_{i,j} = x_k &\Rightarrow l_{i,j}(\overrightarrow{b}) = b_k \\ &\Rightarrow \psi(l_{i,j}) = v_k \end{aligned}$$

  Entonces:

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

- Caso (2):

$$\begin{aligned} l_{i,j} = \overline{x_k} &\Rightarrow l_{i,j}(\overrightarrow{b}) = \overline{b_k} \\ &\Rightarrow \psi(l_{i,j}) = w_k \end{aligned}$$

  Entonces:

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

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

$\boxed{\Rightarrow}$

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:

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

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

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

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

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

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

- Caso (1):

$$\begin{aligned} l_{i,j} = x_k \quad &\Rightarrow \quad l_{i,j}(\overrightarrow{b}) = b_k = 1 \Rightarrow C(v_k) = \text{ BLANCO} \\ &\Rightarrow \quad \psi(l_{i,j}) = v_k \Rightarrow \mu_{i,j}\psi(l_{i,j}) = \underbrace{\mu_{i,j}}_{N}\underbrace{v_k}_{B} \text{ NCP} \end{aligned}$$

- Caso (2):

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

Quedan las bases:

$$C(b_{i,r}) = \begin{cases} \text{AZUL} & \text{si } r = j \\ \text{BLANCO} & \text{si } r \neq j \\ \text{NEGRO} & \text{si } r \neq j \end{cases}$$

Entonces:

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

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