MATEMATICA DISCRETA II-2006
P-NP

1.Definicion :

Un problema de decisión es un problema cuyas unicas respuestas posibles son "SI" o "NO".

Ejemplos:

1) POSITIVIDAD: Dado un numero n, ¿Es $n > 0$?

2) MATCHING: Dado un grafo G, ¿Existe un matching perfecto en G?

3) MATCHING BIPARTITO: Idem que el anterior, pero ahora suponemos que G es bipartito.

4) MATCHING BIPARTITO PESADO: Dado un grafo bipartito G con pesos en los lados y un numero r, ¿existe un matching tal que la suma de los pesos del matching sea menor o igual a r?

5) k-COLOR: Dado un grafo G ¿Existe un coloreo propio con k colores de G?

6) v-flujo: Dado un network N y vertices s, t ¿existe un flujo de s a t con valor v?

7) PRIMO: Dado un numero natural n, ¿Es n primo?

8) COMPUESTO: Dado un numero natural n, ¿Es n compuesto, i.e., no primo?

9) INVERTIBILIDAD: Dada una matriz A $n \times n$, ¿Es A invertible?

10) CONECTITUD: Dado un grafo G, ¿Es G conexo?

2.Definicion :

Dado un problema de decision π, una INSTANCIA de π es un valor particular de los datos de entrada de π.

Por ejemplo, en 1), 7) y 8) arriba, una instancia seria un numero concreto n, por ejemplo, $n = 15$. En 9) seria una matriz concreta, en 2), 5) y 10) seria un grafo concreto, etc.

3.Definicion :

La clase P es la clase de problemas de decisión tal que existe un algoritmo que es polinomial en el tamaño de sus instancias que lo resuelve.

Ejemplos: 1) esta en P. Vimos que 3), 4) y 6) estan en P. En Nuemrico vieron que 9) esta en P, y en Algoritmos II que 10) esta en P (BFS o DFS). No vimos, pero Edmonds probò que 2) esta en P. Hasta hace poco no se sabia, pero en el 2003 se probo que PRIMO (y por lo tanto COMPUESTO) esta en P.

Tambien vimos que 2-COLOR esta en P.

Quedaria por saber, por ejemplo, 3-COLOR.

Observemos que, si bien tenemos algoritmos que resuelven estos problemas, en varios casos tambien tenemos "certificados" para estos problemas. Por ejemplo, si no hay un matching, el teorema de Hall nos da un S tal que $|S| > |\Gamma(S)|$. En este caso, en vez de resolver el problema de matching, si alguien nos da S, solo debemos verificar la ecuacion anterior para saber que no hay matching.

4.Definicion :

Un certificado para "SI" de una instancia I de un problema de decision π es una isntancia C de un problema $\tilde{\pi}$ tal que $\tilde{\pi}(C) =$ "SI" $\Rightarrow \pi(I) =$ "SI". (Esto se llama "verificar el certificado").

(una definicion similar para "NO").

Ejemplos:

1) Un certificado (para "SI") de una instancia A de INVERSIBILIDAD puede ser el par (A, A^{-1}). En este caso, el problema $\tilde{\pi}$ es: "'Dadas dos matrices A y B $n \times n$, ¿es cierto que

$AB = I$? Observar que $\tilde{\pi}(A, B) =$"SI" implica $\pi(A) =$"SI", pero $\tilde{\pi}(A, B) =$"NO" no implica $\pi(A) =$"NO", pues $\tilde{\pi}(A, B) =$"NO" solo dice que B no es la inversa de A, pero A podria si tener inversa.

¿Cual podria ser un certificado para "NO" de este problema? A primera vista, no parece existir uno, pero si uno recuerda la teoria de Algebra lineal, recordara que una matriz es inversible si y solo si para todo $AX = 0 \Rightarrow X = 0$, con lo cual un certificado para "NO" seria un par (A, X) con X un vector columna, y el problema $\tilde{\pi}$ seria ahora: "Dada una matriz A $n \times n$ y un vector $n \times 1$ $X \neq 0$, ¿es cierto que $AX \neq 0$?" Asi, $\tilde{\pi}(A, X) =$"NO" indicaria que $AX = 0$ con $X \neq 0$, y por la caracterizacion arriba indicada de inversibilidad, esto diria que A no es inversible, es decir $\pi(A) =$"NO". Por otro lado, $\tilde{\pi}(A, X) =$"SI" no diria nada: solo sabriamos que X no esta en el nucleo de A, pero no sabemos si existe o no algun otro vector ahi.

En este caso planteado, el certificado para "SI" no es un certificado para "NO" y viceversa, pero, en el caso de INVERSIBILIDAD, existe un certificado que es al mismo tiempo certificado para "SI" y "NO" simultaneamente: el determinante de la matriz. En este caso, solo hay que verificar si el mismo es o no distinto de cero. En general, no tendremos tanta suerte.

2) Un certificado para "SI" para k-COLOR es un coloreo con k colores. El proplema $\tilde{\pi}$ es :"Dado un grafo G y un coloreo de G, ¿Es el coloreo propio?". Observar que $\tilde{\pi}$ tiene complejidad $O(m)$, i.e, hemos reducido un problema que no sabemos cual es su complejidad, a uno en P. El problema, por supuesto, es que alguien debe suministrarnos el coloreo.

5.Definicion :

La clase NP es la clase de problemas de decision para los cuales para toda instancia I del problema, existe un certificado para "SI" tal que existe un algoritmo que verifica el certificado en tiempo polinomial. (polinomial en el tamaño de I)

La clase co-NP se define similarmente, reemplazando "SI" por "NO".

¿de donde viene la notacion? Algunos autores le llaman a NP "VP" por "verificable polinomialmente", lo cual seria mas facil de acordarse, pero el "N" viene de "No determinisco": esto es porque si un problema se puede verificar en tiempo polinomial, entonces se puede "resolver" en tiempo polinomial, *si permitimos computadores NO DETERMINISTICOS*.

Por ejemplo, hemos visto arriba que INVERSIBILIDAD esta en $P \cap co - NP$. (en realidad, sabemos que esta en P). Tambien vimos que k-COLOR esta en NP.

Veamos otro ejemplo:

Un certificado para "SI" de una instancia n de COMPUESTO es un triple (a, b, n) con $1 < a, b < n$, y con $\tilde{\pi}$: "Dados (a, b, n) con $1 < a, b, n$ ¿Es cierto que $ab = n$?

Verificar que $ab = n$ se realiza en tiempo polinomial, por lo tanto COMPUESTO$\in NP$ (y, por lo tanto PRIMO$\in co - NP$).

Observar que, otra vez, si la respuesta es "NO" no sabemos que n es primo, solo que ab no es igual a n, pero n podria factorizarse de otra forma.

Observar que esto nos da un certificado para "NO" de PRIMO. No esta claro cual seria una instancia para "SI" de PRIMO, es decir, no esta claro que PRIMO$\in NP$. En los ́70s, se descubrio un certificado, medio complicado, que prueba que PRIMO$\in NP$. De hecho, como dije antes, hace unos años, se probo que PRIMO$\in P$.

En general, esta claro que $P \subseteq NP$, pues si puedo resolver un algoritmo en tiempo polinomial, simplemente tomo $\tilde{\pi} = \pi$.

6.Pregunta :

¿Existe algun problema en NP que NO este en P? (i.e., ¿Es $P = NP$?).

Respuesta: vale un millon de dolares.

(Se sabe que hay problemas que NO estan en P, pero tampoco estan en NP!)

De los ejemplos que vimos arriba, todos, menos uno, estan en P, asi que no nos sirven. El que queda es k-COLOR, con $k \geq 3$. Veamos otro:

7.Definiciones :

.-Una variable booleana es una variable que solo toma los valores 1 ("True") o 0 ("False").

.-Una expresion booleana es una funcion en variables booleanas.

.- Dadas variables booleanas $x_1, ..., x_n$, un literal es una variable x_i o la negacion de una variable: $\overline{x}_i$.

.- Una disjuncion es una expresion booleana de la forma $\ell_1 \vee ... \vee \ell_k$ donde los ℓ_i son literales.

.- Dadas expresiones booleanas $B_1, ..., B_m$, la conjuncion de ellas es $B_1 \wedge ... \wedge B_m$.

.- Una expresion booleana esta en forma conjunta normal (CNF: conjuntive normal form) si es una conjuncion de disnjunciones.

Ejemplo:

$$(x_1 \vee x_2) \wedge (\overline{x}_1 \vee x_3 \vee x_4) \wedge (x_2 \vee \overline{x}_3) \wedge (x_1 \vee \overline{x}_2 \vee \overline{x}_4)$$

8.Definicion :

El problema "SAT" o problema de satisfacibilidad es: Dada una expresion booleana en forma conjuntiva normal, ¿existe un asignamiento de valores a las variables que la haga verdadera?

Por ejemplo, en la instancia de SAT del ejemplo anterior, la respuesta es "SI" porque podemos tomar $x_i = 1$ para todo i y tenemos que la expresion evalua a:

$$(1 \vee 1) \wedge (0 \vee 1 \vee 1) \wedge (1 \vee 0) \wedge (1 \vee 0 \vee 0) = 1$$

Sin embargo, si esta no fuese una solucion, debriamos seguir buscanco otra. Si hay n variables, esto nos da un total de 2^n posibilidades, por lo que este sistema de fuerza bruta no anda. Otra posibilidad es darse cuenta que si la expresion esta en CNF, con m disjunciones, entonces si queremos encontrar una solucion, cada disjuncion debe ser "true", por lo tanto en realidad debemos resolver un sistema de m ecuaciones en n incongitas...pero el problema es que el sistema no es lineal.

Hasta el momento, nadie sabe como resolver esto en tiempo polinomial, es decir, nadie sabe si $SAT \in P$ o no.

Esta claro que $SAT \in NP$, pues un certificado es simplemente un asignamiento de valores a las variables: dado un tal asignamiento, verificar que la expresion evalua a 1 en el es ciertamente polinomial.

Si esto fuera todo lo que se sabe de P-NP, ciertamente no seria mucho.

Pero existe un concepto clave, que es el de *reduccion*:

9.Definicion: Reduccion Polinomial :

Dados dos problemas π_1 y π_2, se dice que π_1 es reducible polinomialmente a π_2 ($\pi_1 \leq_P \pi_2$) si existe un algoritmo polinomial que, tomando como input ina instancia I_1 de π_1, produce como output una instancia I_2 de π_2 con la propiedad de que:

$$\pi_1(I_1) = \text{"SI"} \iff \pi_2(I_2) = \text{"SI"}$$

La idea es la siguiente: $\pi_1 \leq_P \pi_2$ si es "mas facil" resolver π_1 que π_2, en el sentido que, si alguien tiene un algoritmo que puede resolver π_2, entonces podemos, al querer calcular $\pi_1(I_1)$, en vez de ello, usar la transformacion para obtener I_2, darle I_2 como input a π_2, y conociendo el resultado $\pi_2(I_2)$, saber el resultado de $\pi_1(I_1)$.

10.Corolario :

$\pi_1 \leq_P \pi_2 \in P \Rightarrow \pi_1 \in P$

Prueba:

Pues si tengo un algoritmo polinomial para reducir una instancia de π_1 a una de π_2, y tengo un algoritmo polinomial para resolver π_2, entonces el algoritmo: "reducir a π_2 y luego resolver π_2" es polinomial. QED.

Ejercicio: Probar que $\leq_P$ es transitiva.

11.Teorema :

$$k - \text{COLOR} \leq_P \text{SAT}$$

Prueba:

Debemos dar un algoritmo (polinomial) que, dada una instancia de k-COLOR (i.e., un grafo G) nos produsca un ainstancia de SAT (i.e., una expresion booleana B en CNF) tal que $\chi(G) \leq k$ si y solo si B es satisfacible (i.e., existe un asignamiento de variables que la vuelve verdadera).

Sea G el grafo, y n el numero de vertices. Supongamos que los vertices son $1, 2, ..., n$. Tomemos nk variables booleanas $x_{i,j}$ con $i = 1, ..., n$, $j = 1, ..k$. Definamos las siguientes expresiones booleanas:

$$\begin{aligned}
A_i &= x_{i,1} \vee x_{i,2} \vee ... \vee x_{i,k} \\
A &= A_1 \wedge ... \wedge A_n \\
D_i &= \bigwedge_{j<r} (\overline{x}_{i,j} \vee \overline{x}_{i,r}) \\
D &= D_1 \wedge ... \wedge D_n \\
F &= \bigwedge_{ih \in E(G)} \bigwedge_{j=1}^{k} (\overline{x}_{i,j} \vee \overline{x}_{h,j}) \\
B &= A \wedge D \wedge F
\end{aligned}$$

B esta en CNF y el algoritmo que construye B a partir de G es claramente polinomial.

Veamo ahora el si y solo si:

.-$\chi(G) \leq k \Rightarrow B$ es satisfacible:

Sea c un coloreo de G con a lo sumo k colores. (asumamos que los colores son los numeros $1, ..., k$) Definamos el asignamiento de valores dado por :

$$x^o_{i,j} = \begin{cases} 1 & \text{si } c(i) = j \\ 0 & \text{si no} \end{cases}$$

Veamos que $B(\vec{x}^o) = 1$, donde $\vec{x}^o = (x^o_{i,j})_{i,j}$. Para ello, veamos que tanto A com D como F, evaluados en los $x^o_{i,j}$ dan 1:

1) Puesto que c es un coloreo con a lo sumo k colores, entonces para todo i hay algun $j \leq k$ tal que $c(i) = j$. Asi, $\forall i \exists j \leq k$ tal que $x^o_{i,j} = 1$. Pero entonces $x^o_{i,1} \vee x^o_{i,2} \vee .... \vee x^o_{i,k} = 1$, es decir, $A_i(\vec{x}^o) = 1$ para todo i, es decir, $A(\vec{x}^o) = 1$.

2) Como c es ua funcion, para todo i hay un UNICO j tal que $c(i) = j$. Es decir, NO existen $j < r$ tal que $c(i) = j$ y $c(i) = r$. Por lo tanto, $\forall i \; \nexists j < r$ tal que $x^o_{i,j} = 1$ y $x^o_{i,r} = 1$, lo que equivale a decir que $\forall i \forall j < r$, o bien $x^o_{i,j} = 0$, o bien $x^o_{i,r} = 0$. Como $x^o_{i,j} = 0$ es equivalente a $\overline{x}^o_{i,j} = 1$, tenemos que $\forall i \forall j < r$ vale que $\overline{x}^o_{i,j} \vee \overline{x}^o_{i,r} = 1$ i.e., $\bigwedge_i \bigwedge_{j<r} (\overline{x}^o_{i,j} \vee \overline{x}^o_{i,r}) = 1$, i.e., $D(\vec{x}^o) = 1$.

3) Como c es coloreo propio, para todo lado ih tenemos que $c(i) \neq c(h)$. Esto dice que para todo j, o bien $c(i) \neq j$, o bien $c(h) \neq j$. Es decir, $\forall ih \in E(H), \forall j$ vale $x^o_{i,j} = 0$ o bien $x^o_{h,j} = 0$. Como antes, esto dice que $\bigwedge_{ih \in E(g)} \bigwedge_j (\overline{x}^o_{i,j} \vee \overline{x}^o_{h,j}) = 1$, i.e., $F(\vec{x}^o) = 1$.

Por lo tanto, hemos probado la implicación $\chi(G) \leq k \Rightarrow B$ es satisfacible. Veamos la vuelta:

B es satisfacible $\Rightarrow \chi(G) \leq k$. Ahora tenemos un asignamiento tal que $B(\vec{x}^o) = 1$. Por la misma cuenta de antes, $D(\vec{x}^o) = 1$ implica que para cada i hay a lo sumo un j tal que $x^o_{i,j} = 1$. Y tambien, $A(\vec{x}^o) = 1$ implica que para todo i hay al menos un j tal que $x^o_{i,j} = 1$. Por lo tanto, $\forall i \exists ! j$ tal que $x^o_{i,j} = 1$. Definamos $c(i)$ como el unico j tal que $x^o_{i,j} = 1$. ESto esta bien defiido, son a lo sumo k colores, y como $F(\vec{x}^o) = 1$, por la cuenta anterior, obtenemos que c es un coloreo propio. QED.

"De la misma forma" se puede probar mas todavia:

12.Teorema :

(Cook, 1972)

$$\pi \in NP \Rightarrow \pi \leq_P \text{SAT}$$

Prueba:

No la veremos. Es mas o menos como la anterior, i.e., consiste en expresar el problema con una expresion booleana. El problema es ver que todo problema en NP puede expresarse de esa forma.

13.Corolario :

$$SAT \in P \Rightarrow P = NP$$

14.Definición :

Un problema τ es NP-completo (o tambien se dice que esta en NPC si $\tau \in$NP y $\pi \in$NP$\Rightarrow \pi \leq_P$SAT.

Por lo tanto, el teorema de Cook se puede escribir como "SAT$\in$NPC". Veremos que SAT no es el unico problema NPcompleto. En general, si τ es NP-completo, yπ es un problema NP con $\tau \leq_P \pi$, entonces π tambien es NP completo.

15.Definicion :

El problema 3-SAT es como SAT, pero se especifica que la expresion booleana debe tener exactamente 3 literales en cada disjunción.

El problema (≤ 3)-SAT es como SAT, pero la expresion booleana tiene que tener a lo sumo tres literales en cada disjunción.

Ejercicio: probar que (≤ 3)-SAT$\leq_P$ 3-SAT.

16.Teorema :

3-SAT es NP completo.

Prueba:

Por el ejercicio, basta probar que (≤ 3)-SAT es NP-completo. Para ello, veremos que SAT$\leq_P$ (≤ 3)-SAT. Es decir, dada una instancia de SAT (es decir, una expresion booleana B en CNF), debemos construir polinomialmente una instancia de (≤ 3)-SAT (i.e., una expresion booleana $\tilde{B}$ en CNF, tal que $\tilde{B}$ tenga a lo sumo 3 literales en cada disjunción), de tal forma que B sea satisfacible si y solo si $\tilde{B}$ sea satisfacible.

Como $B = D_1 \wedge ... \wedge D_m$ para algunas disjunciones D_i, simplemente daremos un algoritmo para transformar cada disjunción D_i en una expresion E_i que será conjunción de disjunciones, cada una de las cuales con a lo sumo 3 literales. Entonces $\tilde{B} := E_1 \wedge ... \wedge E_m$ estara en CNF con a lo sumo tres disjunciones en cada disjunción. Mas aun, lo haremos de forma tal de tener compatibilidad de asignamiento entre las D_i's: es decir, si las variables sde B son $\vec{x} = (x_1, ..., x_n)$, entonces cada E_i dependera de variables $(\vec{x}, \vec{y_i}) = (x_1, ..., x_n, y_{1,i}, ..., y_{n,i})$. Probaremos que existe un asignamiento $\vec{x}^o = (x_1^o, ..., x_n^o)$ tal que $D_i(\vec{x}^o) = 1$ si y solo si existe un asignamiento de la forma $(\vec{x}^o, \vec{y}_i^o)$ tal que $E_i(\vec{x}^o, \vec{y}_i^o) = 1$.

Asi, existe $\vec{x}^o$ tal que $B(\vec{x}^0) = 1$ si y solo si $D_i(\vec{x}^o) = 1 \forall i$ si y solo si $\exists \vec{y}_1^0, ..., \vec{y}_m^0$ con $E_i(\vec{x}^0, \vec{y}_i^o) = 1$ si y solo si $\tilde{B}(\vec{x}^o, \vec{y}_1^o, ..., \vec{y}_m^o) = 1$.

Por lo tanto, nos podemos concentrar en el caso de una disjunción D. (asi eliminamos el subindice i).

Si D tiene 3 o menos literales, no hacemos nada, es decir, tomamos $E = D$.

Supongamos que D tiene $k \geq 4$ disjunciones: $D = \ell_1 \vee ... \vee \ell_k$. Introducimos nuevas variables $y_1, ..., y_{k-3}$.

Definimos E como la conjunción de las siguientes disjunciones:

$$\begin{aligned} E = \quad & (\ell_1 \vee \ell_2 \vee y_1) \wedge (\overline{y}_1 \vee y_2 \vee \ell_3) \wedge (\overline{y}_2 \vee y_3 \vee \ell_4) \wedge \\ & \wedge .... \wedge (\overline{y}_{k-4} \vee y_{k-3} \vee \ell_{k-2}) \wedge (\overline{y}_{k-3} \vee \ell_{k-1} \vee \ell_k) \end{aligned}$$

Supongamos ahora que existe $\vec{x}^o, \vec{y}^o)$ tal que $E(\vec{x}^o, \vec{y}^o) = 1$. Probemos que $D(\vec{x}^o) = 1$.

Supongamos que esto no fuera cierto, es decir, que $D(\vec{x}^o) = 0$. Como D es una disjuncion, tenemos que $\ell_1^o \vee ... \vee \ell_k^o = 0$, es decir, $\ell_j^o = 0$ para todo j.

Reeemplazando esto en E, tenemos que:

$$\begin{aligned} 1 = E(\vec{x}^o, \vec{y}^o) = \quad & (\ell_1^o \vee \ell_2^o \vee y_1^o) \wedge (\overline{y}_1^o \vee y_2^o \vee \ell_3^o) \wedge (\overline{y}_2^o \vee y_3^o \vee \ell_4^o) \wedge \\ & \wedge .... \wedge (\overline{y}_{k-4}^o \vee y_{k-3}^o \vee \ell_{k-2}^o) \wedge (\overline{y}_{k-3}^o \vee \ell_{k-1}^o \vee \ell_k^o) \\ = \quad & (y_1^0) \wedge (\overline{y}_1^o \vee y_2^o) \wedge (\overline{y}_2^o \vee y_3^o) \wedge \\ & \wedge .... \wedge (\overline{y}_{k-4}^o \vee y_{k-3}^o) \wedge (\overline{y}_{k-3}^o) \end{aligned}$$

Por lo tanto, por la primera disjuncion, debemos tener que $y_1^o = 1$, es decir, $\overline{y}_1^o = 0$. Pero entonces, como $\overline{y}_1^o \vee y_2^o = 1$ por la segunda, debemos tener que $y_2^o = 1$. Continuando de esta forma, tendemos que $y_r^o = 1$ para todo r. Pero la ultima disjuncion dice que $\overline{y}_{k-3}^o = 1$, es decir, $y_{k-3} = 0$, lo cual es una contradicción.

Por lo tanto, hemos probado una implicación.

Veamos la otra, es decir, supongamos que existe $\vec{x}^o$ tal que $D(\vec{x}^o) = 1$.

Entonces alguno de los ℓ_j^o debe ser 1.

Sea r el primer indice tal que $\ell_r^o = 1$.

Definamos:

$$y_j^o = \begin{cases} 1 & \text{si } j \leq r-2 \\ 0 & \text{si } j \geq r-1 \end{cases}$$

Entonces, la primera disjunción $\ell_1^o \vee \ell_2^o \vee y_1^o = 1$ pues $y_1^o = 1$. La ultima disjuncion $\overline{y}_{k-3}^o \vee \ell_{k-1}^o \vee \ell_k^o = 1$ pues $y_{k-3}^o = 0$ y por lo tanto $\overline{y}_{k-3} = 1$. En cuanto a las disjunciones intermedias:

$$\overline{y}_{j-1}^o \vee y_j^o \vee \ell_{j+1}^o = \begin{cases} 0 \vee 1 \vee 0 = 1 & \text{si } j \leq r-2 \text{ pues } y_j^o = 1 \\ 0 \vee 0 \vee 1 = 1 & \text{si } j = r-1 \text{ pues } \ell_{j+1}^o = \ell_r^o = 1 \text{ si } j = r-1. \\ 1 \vee 0 \vee \ell_{j+1}^o = 1 & \text{si } j \geq r \text{ pues } \overline{y}_{j-1}^o = 1 \text{ en este caso} \end{cases}$$

Asi, cada disjuncion es 1 y $E(\vec{x}^o, \vec{y}^o) = 1$. QED.

17. Teorema :

3-COLOR es NP-completo.

Prueba:

Por lo anterior, basta ver que 3-SAT$\leq_P$3-COLOR.

Es decir, dada una instancia de 3-SAT; i.e., una expresion booleana B en CNF con exactamente 3 literales en cada disjunción; debemos crear una instancia de 3-COLOR; i.e., un grafo G; tal que B sea satisfacible si y solo si $\chi(G) \leq 3$.

Supongamos que las variables de B son $x_1, ..., x_n$ y que $B = D_1 \vee ... \vee D_m$, con las disjunciones $D_i = \ell_1(i) \vee \ell_2(i) \vee \ell_3(i)$.

Definamos el grafo G: sus vertices seran :

$$V(G) = \{s, t\} \cup \{v_j\}_{j=1}^n \cup \{w_j\}_{j=1}^n \cup \{e_k(i)\}_{\substack{k=1,2,3 \\ i=1,..,m}} \cup \{a_k(i)\}_{\substack{k=1,2,3 \\ i=1,..,m}}$$

es decir, un total de $2 + n + n + 3m + 3m = 2 + 2n + 6m$ vertices.

Para describir los lados, primero necesitamos una definicion: dado un literal ℓ de B (por lo tanto ℓ es alguna variable x_j o alguna negacion de alguna variable, $\overline{x}_j$), definimos *el vertice correspondiente a* ℓ como:

$$v(\ell) = \begin{cases} v_j & \text{si } \ell = x_j \\ w_j & \text{si } \ell = \overline{x}_j \end{cases}$$

Los lados serán:

$$\begin{aligned} E(G) = &\{st\} \cup \{tv_j\}_{j=1}^n \cup \{tw_j\}_{j=1}^n \cup \{v_j w_j\}_{j=1}^n \\ &\cup \{e_k(i)a_k(i)\}_{\substack{k=1,2,3 \\ i=1,..,m}} \cup \{a_1(i)a_2(i), a_2(i)a_3(i), a_3(i)a_1(i)\}_{i=1,..,m} \cup \\ &\cup \{se_k(i)\}_{\substack{k=1,2,3 \\ i=1,..,m}} \cup \{e_k(i)v(\ell_k(i))\}_{\substack{k=1,2,3 \\ i=1,..,m}} \end{aligned}$$

Es decir, un total de $1 + n + n + n + 3m + 3m + 3m + 3m = 1 + 3n + 12m$ lados.

(la primera linea de lados define un abanico de triangulos con bases $v_j w_j$ y vertice comun superior t, ademas del lado st. La segunda linea define m "garras" formadas por un triangulo de vertices "adentro" de la garra y "uñas" externas e_k; unidas cada una de ellas a un vertice del triangulo de adentro. La tercera linea dice que los extremos de las garras se unen a s, y ademas cada uno se une a un vertice correspondiente a un literal de la disjunción D_i).

Antes de poder seguir, debemos probar un:

Lemma Interno:*Sea c un coloreo (quizas no propio) de G y $\vec{x}^o$ un asignamiento de las valores a las variables de B. Supongamos que $c(v_j) = x_j^o$ y que $c(w_j) = \overline{x}_j^o$. Sea ℓ un literal cualquiera. Entonces $c(v(\ell)) = \ell^o$*

Prueba:

Como ℓ es un literal, es una variable o la negación de una variable. Supongamos que $\ell = x_j$. Entonces, por un lado $v(\ell) = v_j$, y por otro, $\ell^o = x_j^o$. Asi:

$$c(v(\ell)) = c(v_j) = x_j^o = \ell^o$$

Supongamos ahora que $\ell = \overline{x}_j$. Entonces, por un lado, $v(\ell) = w_j$, y por el otro $\ell^o = \overline{x}_j^o$. Asi:

$$c(v(\ell)) = c(w_j) = \overline{x}_j^o = \ell^o$$

y hemos probado el lema interno. †

Probemos ahora que $\exists \vec{x}^o$ con $B(\vec{x}^o) = 1$ si y solo si $\chi(G) \leq 3$. (en realidad, debera ser $\chi(G) = 3$ pues al tener triangulos sabemos que no es bipartito).

($\Rightarrow$): Sea $\vec{x}^o$ tal que $B(\vec{x}^o) = 1$. Coloreemos $c(v_j) = x_j^o$ y $c(w_j) = \overline{x}_j^o$. En particular, el color de v_j es 1 sii el de w_j es cero, por lo tanto los lados $v_j w_j$ no crean problemas. Si coloreamos t con el color 2, entonces los lados tv_j y tw_j no crean problemas. Coloreando s con el color 1, el lado st no creara problemas.

Como $B(\vec{x}^o) = 1$, entonces para cada i tenemos $D_i(\vec{x}^o) = 1$. Como $D(\vec{x}^o) = \ell_1^o(i) \vee \ell_2^o(i) \vee \ell_3^o(i)$, entonces existe el menos un $k = k_i$ tal que $\ell_k^o(i) = 1$. (si hay mas de uno me quedo con uno solo).

Entonces, por el lema interno, $c(v(\ell_k(i)) = 1$, por lo que si definimos $c(e_k(i)) = 0$, los lados $e_k(i)v(\ell_k(i))$ no crean problemas. Coloreamos los otros dos extremos de la garra i-esima con el color 2. Como $c(v(\ell_j(i))) = 0$ o 1, esto hace que los lados $e_j(i)v(\ell_k(i))$ con $j \neq k_i$ no crean problemas. Finalmente, coloreamos $a_k(i)$ con el color 2, lo cual no causa problema con el lado $a_k(i)e_k(i)$ pues un extremo tien el color 2 y el otro el color 0, y coloreamos los dos vertices restantes con los colores 1 y 0 (uno con el 0 y uno con el 1). Esto hace que el triangulo interno de la garra no cause problemas, y los lados $e_j(i)a_j(i)$ con $j \neq k_i$ no crean problemaspues los $e_j(i)$ tienen color 2 y los $a_j(i)$ color 0 o 1. Asi, hemos coloreado todo el grafo con 3 colores.‡

Veamos la vuelta:

($\Leftarrow$): Sea ahora un coloreo c de G con 3 colores. Llamemos 1 al color de s, 2 al color de t y 0 al color restante. Como tv_j sonm lados, entonces el color de v_j no puede ser 2, y por lo tanto, es 0 o 1, para todo j. Por lo tanto, si definimos $x_j^o = c(v_j)$, esto es un asignamiento booleano. Como $tw_j \in E$, tambien tenemos que $c(w_j) \in \{0, 1\}$, y como los $v_j w_j \in E$, debe ser $c(w_j) \neq c(v_j)$, por lo tanto, $c(w_j) = \overline{c(v_j)} = \overline{x}_j^o$. Por el lema interno, entonces $c(v(\ell)) = \ell^o$ para cualquier literal ℓ.

Tomemos un i cualquiera. Como el triangulo $\{a_1(i)a_2(i), a_2(i)a_3(i), a_3(i)a_1(i)\}$ debe tener los tres colores, existe un k tal que $c(a_k(i)) = 2$. Por lo tanto:

$$\left.\begin{array}{ccc} e_k(i)a_k(i) \in E & \Rightarrow & c(e_k(i)) \neq 2 \\ e_k(i)s \in E & \Rightarrow & c(e_k(i)) \neq 1 \end{array}\right\} \Rightarrow c(e_k(i)) = 0$$

y entonces:

$$\left.\begin{array}{ccc} e_k(i)v(\ell_k(i)) \in E & \Rightarrow & c(v(\ell_k(i))) \neq 0 \\ v(\ell_k(i))t \in E & \Rightarrow & c(v(\ell_k(i))) \neq 2 \end{array}\right\} \Rightarrow c(v(\ell_k(i))) = 1 \Rightarrow \ell_k^o(i) = 1$$

Pero $\ell_k^o(i) = 1$ implica que $D(\vec{x}^o) = 1$. Como esto vale para todo i, entonces $B(\vec{x}^o) = 1$. QED.