



Tópicos Avanzados en Teoría de Computación

Demostrar que pertenece a DP

$$\text{clique-number}(G) = \max\{k \mid G \text{ tiene un clique de tamaño } k\}$$

$$\text{exact-CLIQUE} = \{(G, k) \mid \text{clique-number}(G) = k\}$$

Para demostrar que **exact-CLIQUE** (EC) pertenece a la clase *DP* debemos escribir la definición del lenguaje como la diferencia de dos lenguajes que están en NP.

$$\text{CLIQUE} = \{(G, k) \mid G \text{ tiene un clique de tamaño } \geq k\}$$

CLIQUE $\in$ NP, ¿por qué?

- Máquina no determinista recibe $w = (G, k)$, G un Grafo, y $k \in \mathbb{Z}$
- Se genera un subgrafo con k nodos a partir de G
- Se comprueba si este subgrafo corresponde a un clique (Tiempo polinomial)
- Si corresponde a un clique, entonces llegamos a estado final, en caso contrario rechaza.

$$\text{EC} = \{(G, k) \mid G \text{ tiene un clique de tamaño } \geq k \wedge G \text{ No tiene un clique de tamaño } \geq k + 1\}$$

Entonces EC queda escrito de la forma $\text{EC} = L1/L2$ con $L1$ y $L2$ en NP

Con esto demostramos que **exact-CLIQUE** $\in$ **DP**



Tópicos Avanzados en Teoría de Computación

Demostrar que exact-CLIQUE es DP-Hard

Para ello utilizaremos la noción de que el problema SAT-UNSAT es DP-Completo, y se reducirá SAT-UNSAT a exact-CLIQUE, para demostrar que EC es DP-Hard y por consiguiente DP-Completo.

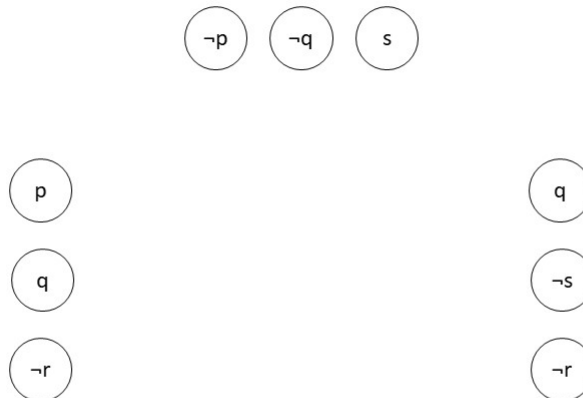
$$\text{SAT-UNSAT} = \{(\gamma, \phi) \mid \gamma, \phi \text{ son formulas en CNF, } \gamma \text{ es satisfacible y } \phi \text{ no es satisfacible} \}$$

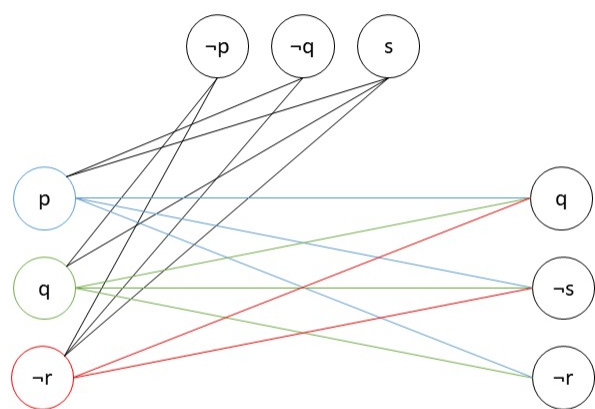
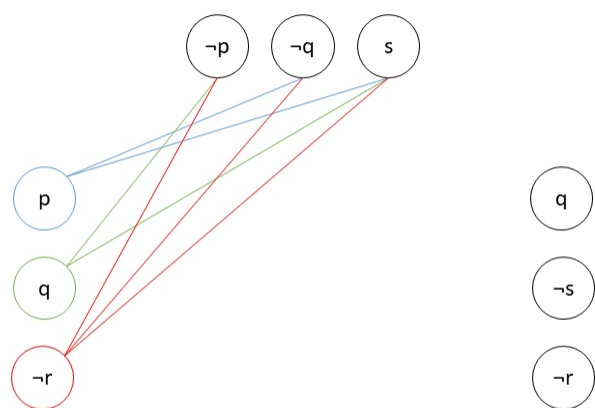
2.1 Reducción CNF-SAT a CLIQUE

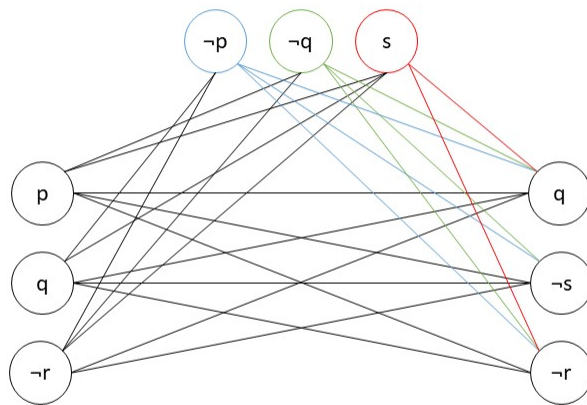
sea $\gamma = (p \vee q \vee \neg r) \wedge (\neg p \vee \neg q \vee s) \wedge (q \vee \neg s \vee \neg r)$

Para crear el grafo a partir de γ (Denotado como G_γ), se hará lo siguiente:

- Por cada clausula en γ creamos un "cluster" en G
- Cada "cluster" contiene nodos con etiquetas de los literales correspondientes a la clausula que lo genera
- Entre cada par de "clusters" se genera una arista que una, cada uno de los nodos de un cluster con todos los nodos del otro. A excepcion de los casos donde la arista sea de la forma $(x, \bar{x})$







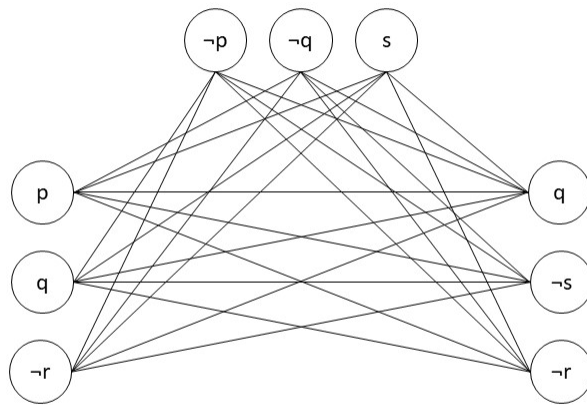
Valuación que satisface:

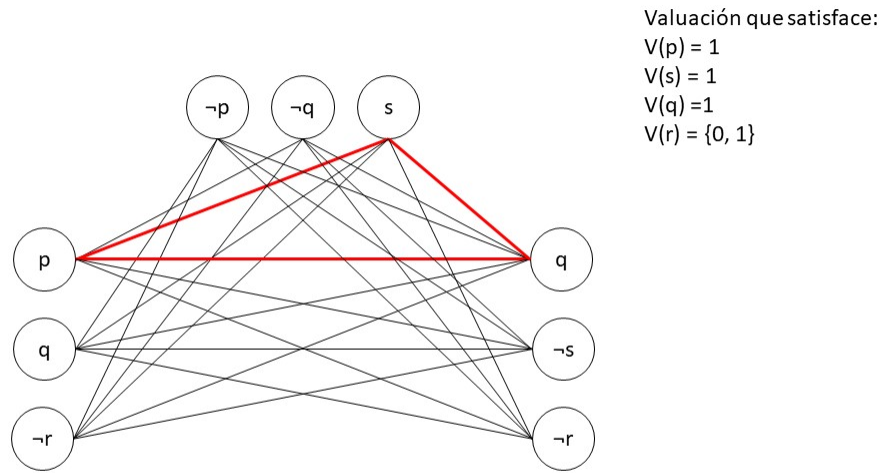
$$V(p) = 1$$

$$V(s) = 1$$

$$V(q) = 1$$

$$V(r) = \{0, 1\}$$





Implicancias de la construcción

- Para cada par de nodos unidos en G_γ , ambos literales pueden ser evaluados de forma tal que sean verdaderos al mismo tiempo.
- Si dos literales $\in \gamma$, que no pertenecen a la misma clausula, pueden ser evaluadas verdaderas a la vez, entonces existe una arista en el grafo G_γ .
- La construcción de este grafo toma tiempo polinomial.

Finalmente notemos que si γ es satisfacible, entonces existe un clique de tamaño igual a la cantidad de clausulas en γ .

De igual manera si el grafo G_γ , tiene un clique de tamaño igual a la cantidad de clausulas en γ , entonces existe una valuación la cual hace verdadera a γ . (Y corresponden con los literales de los nodos que forman el clique)

2.2 Producto cruz entre grafos

Sea $G_1 = (N_1, A_1)$ y $G_2 = (N_2, A_2)$, grafos cualquiera

$$G_1 \times G_2 = (N_1 \times N_2, A)$$

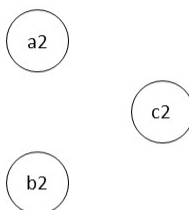
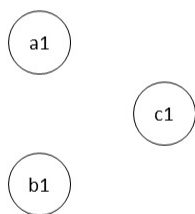
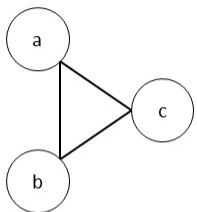
Donde A se construye de la siguiente manera:

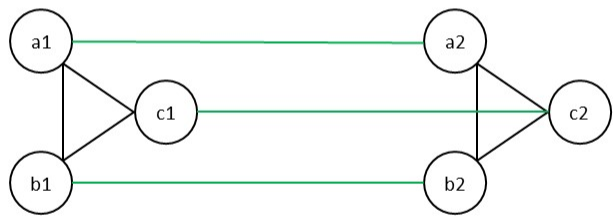
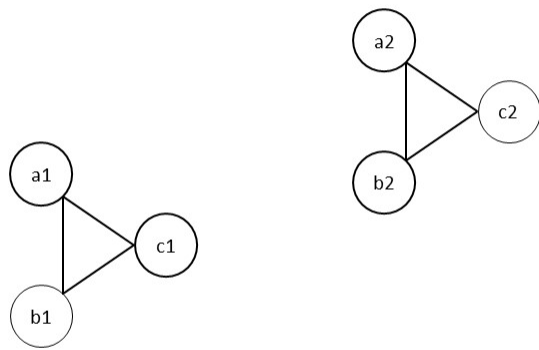
$((a_1, a_2), (b_1, b_2)) \in A$ si se cumple alguna de las siguientes condiciones:

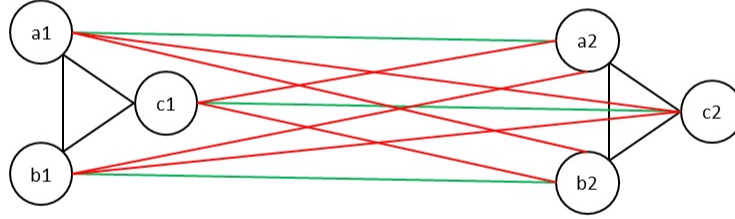
- $(a_1, b_1) \in A_1$ y $(a_2, b_2) \in A_2$
- $a_1 = b_1$ y $(a_2, b_2) \in A_2$

- $(a_1, b_1) \in A_1$ y $a_2 = b_2$

Ejemplo:







Notemos que sucede con el tamaño de los cliques en el grafo resultante. Si G_1 poseía un clique de tamaño k_1 y G_2 tenía un clique de tamaño k_2 , el resultado del producto cruz, nos entregará un grafo que contenga un clique de tamaño igual a $k_1 * k_2$

2.3 Reduccion de SAT-UNSAT a exact-CLIQUE

Dado entonces dos formulas en CNF γ y ϕ , se busca construir un grafo tal que.

$$(\gamma, \phi) \in \text{SAT-UNSAT} \iff (G_{(\gamma, \phi)}, k_{(\gamma, \phi)}) \in \text{exact-CLIQUE}$$

La construcción es la siguiente: Suponiendo que la cantidad de clausulas en $\gamma = m$ y que la cantidad de clausulas en $\phi = n$

Recordando que la notación K_l simboliza un grafo **completo** con l nodos

$$G_{(\gamma, \phi)} = (G_\gamma \cup K_{m-1}) \times (G_\phi \cup K_{n-1})$$

$$K_{(\gamma, \phi)} = m * (n - 1)$$

Explicaciones e implicancias

- La union disjunta de los grafos completos (K) con los grafos generados (G), es para asegurar de que los grafos finales antes de realizar el producto cruz tengan un clique de tamaño al menos $(m - 1)$ para el case de γ y $(n - 1)$ para el caso de ϕ .
- Esto, para que cuando se realice el producto cruz entre estos grafos se obtengan los siguientes posibles resultados.
 - Si **ambas** son satisfacibles: entonces el tamaño del clique final del grafo sería de tamaño $m * n$, puesto que es la cantidad de clausulas en cada una de las formulas

- Si **solo** γ es satisfacible: el tamaño del clique final sería $\mathbf{m} * (\mathbf{n} - \mathbf{1})$, pues corresponde con la cantidad de clausulas de γ , y $n - 1$ es la cantidad de nodos del grafo completo que agregamos K_{n-1} .
- Si **solo** ϕ es satisfacible: el tamaño del clique final sería $(\mathbf{m} - \mathbf{1}) * \mathbf{n}$, corresponde con la cantidad de clausulas de ϕ y la cantidad de nodos en el grafo completo que agregamos K_{m-1}
- Si **ninguna** es satisfacible: el tamaño del clique final corresponde a $(\mathbf{m} - \mathbf{1}) * (\mathbf{n} - \mathbf{1})$, que sería el producto de la cantidad de nodos de los dos grafos completos que agregamos.

Con lo anterior demostramos que, pudimos construir un grafo a partir de 2 formulas tales que

$$(\gamma, \phi) \in \text{SAT-UNSAT} \iff (G_{(\gamma, \phi)}, k_{(\gamma, \phi)}) \in \text{exact-CLIQUE}$$

Finalmente y por lo tanto, se demostró tambien que exact-CLIQUE es DP-Hard y por consiguiente DP-Completo.



Tópicos Avanzados en Teoría de Computación

Ejemplo:

- Sea γ de la forma $C1 \wedge C2$
- Sea ϕ de la forma $C3 \wedge C4 \wedge C5$
- Entonces $m = 2$
- Entonces $n = 3$
- $k = m * (n - 1) = 4$

Construcción del grafo (Vamos a asumir que γ es satisfacible y ϕ no lo es, para hacer mas sencilla la explicación del ejemplo):

