

¿Cuál es la idea central en la técnica mostrada?

Sea (G, k) una entrada de GIS con $k \geq 1$

- ▶ Sabemos que $(G, k) \in \text{GIS}$ si y sólo si $|I^1(G, k)| \geq 1$

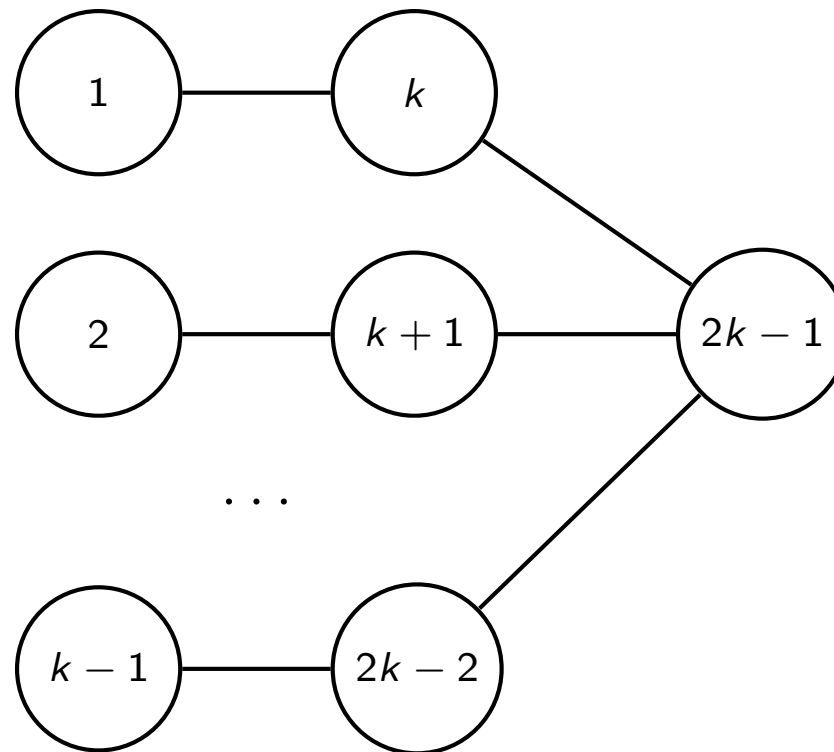
Dado que $|I^1(G, k)| = \#\text{LIS}(G, k) - \#\text{LIS}(G, k - 1)$, podemos intentar utilizar un FPRAS para $\#\text{LIS}$ para determinar si $(G, k) \in \text{GIS}$

- ▶ Generamos así un algoritmo aleatorizado de tiempo polinomial para GIS

El éxito de este enfoque depende de qué tan grande sea el valor de $\#\text{LIS}(G, k) - \#\text{LIS}(G, k - 1)$

¿Qué tan grande es $\#LIS(H, k) - \#LIS(H, k - 1)$?

Considere el siguiente grafo H :



En este caso tenemos que $|I^1(H, k)| = \#LIS(H, k) - \#LIS(H, k - 1) = 1$

► Además, $\#LIS(H, k - 1) \geq 3^{k-1}$

Amplificando la diferencia

La diferencia entre $\#LIS(G, k)$ y $\#LIS(G, k - 1)$ puede ser pequeña.

- ▶ La utilización de un FPRAS para $\#LIS$ no nos garantiza la existencia de un algoritmo aleatorizado de tiempo polinomial para GIS
 - ▶ No podemos acotar superiormente el error de algoritmo

Para solucionar este problema *amplificamos* $\#LIS(G, k) - \#LIS(G, k - 1)$

Amplificando la diferencia

De manera más precisa, de la definición de G^r tenemos:

$\#LIS(G, k) - \#LIS(G, k - 1) > 0$ si y sólo si

$$\#LIS(G^r, k \cdot r) - \#LIS(G^r, (k - 1) \cdot r) > 0$$

Así, usamos la diferencia más grande $\#LIS(G^r, k \cdot r) - \#LIS(G^r, (k - 1) \cdot r)$ para verificar si $\#LIS(G, k) - \#LIS(G, k - 1) > 0$

¿Cuánto amplificando la diferencia?

Suponiendo que G tiene n nodos y tomando $r = n + 3$ obtenemos:

$$\begin{aligned}\#LIS(G^r, k \cdot r) - \#LIS(G^r, (k-1) \cdot r) &\geq (2^r - 1)^k - 2^n \cdot (2^r - 1)^{k-1} \\ &= (2^{n+3} - (2^n + 1)) \cdot (2^{n+3} - 1)^{k-1} \\ &\geq (2^{n+3} - 2^{n+1}) \cdot (2^{n+3} - 1)^{k-1} \\ &= 3 \cdot 2^{n+1} \cdot (2^{n+3} - 1)^{k-1}\end{aligned}$$

¡Podemos entonces verificar si $\#LIS(G, k) - \#LIS(G, k-1) > 0$ considerando una brecha de tamaño exponencial en n !

Una tercera aplicación de la técnica

Sea $G = (N, A)$ un grafo (dirigido) sin loops.

Decimos que $(v_1, \dots, v_n)$ es un ciclo simple en G si:

1. $n \geq 2$
2. $(v_i, v_{i+1}) \in A$ para cada $1 \leq i \leq n-1$, y $(v_n, v_1) \in A$
3. $v_i \neq v_j$ para cada $1 \leq i < j \leq n$

Un ciclo simple $(v_1, \dots, v_n)$ con n vértices puede ser visto como un conjunto con n arcos:

$$\{(v_1, v_2), \dots, (v_{n-1}, v_n), (v_n, v_1)\}$$

En este capítulo utilizamos esta representación de los ciclos simples, y decimos que el largo de este ciclo es n

Una tercera aplicación de la técnica

Sea $\#CicloSimple$ una función que, dado un grafo G , retorna el número de ciclos simples en G

Ejercicio

Muestre que $L_{\#CicloSimple} \in PTIME$

Teorema

Si existe un FPRAS para $\#CicloSimple$, entonces $NP \subseteq BPP$

#CicloSimple no admite un FPRAS: demostración

Sea $\text{HAM} = \{G \mid G \text{ tiene un ciclo hamiltoniano}\}$

Vamos a demostrar que si existe un FPRAS para #CicloSimple, entonces $\text{HAM} \in \text{BPP}$

- ▶ Dado que HAM es NP-completo, se concluye que $\text{NP} \subseteq \text{BPP}$

Sea $G = (N, A)$ un grafo con $|N| = n$ y $n \geq 2$

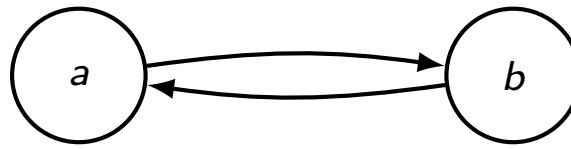
- ▶ G es una entrada de HAM

¿Cómo construimos el grafo aumentado?

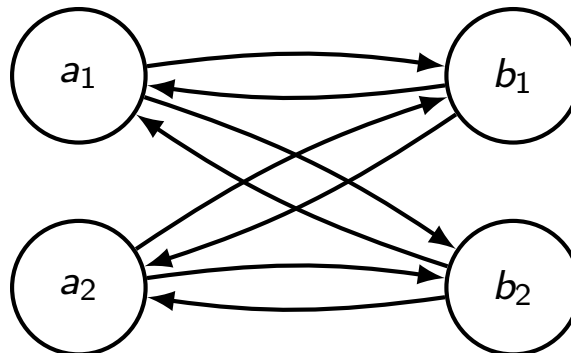
A partir de un ciclo simple con ℓ nodos esperamos generar $2^{\ell \cdot r}$ ciclos simples en el grafo aumentado

- Primer intento: construir G^r como para el caso de $\#LIS$

Considere el siguiente grafo H :



En este caso H^2 es el siguiente grafo:

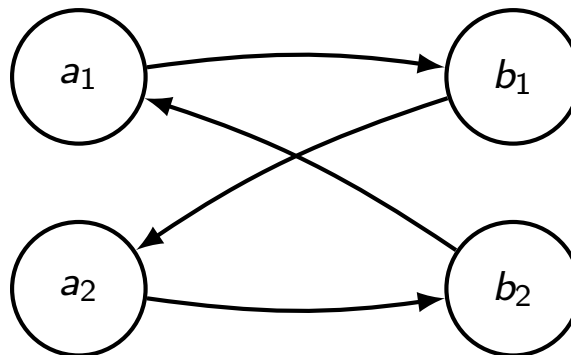


¿Cómo construimos el grafo aumentado?

En H tenemos un ciclo simple y esperamos tener 2^2 ciclos simples en H^2

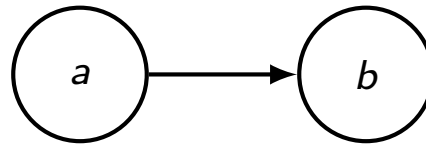
- ▶ (a_1, b_1) , (a_1, b_2) , (a_2, b_1) y (a_2, b_2)

El problema es que H^2 contiene otros ciclos simples:

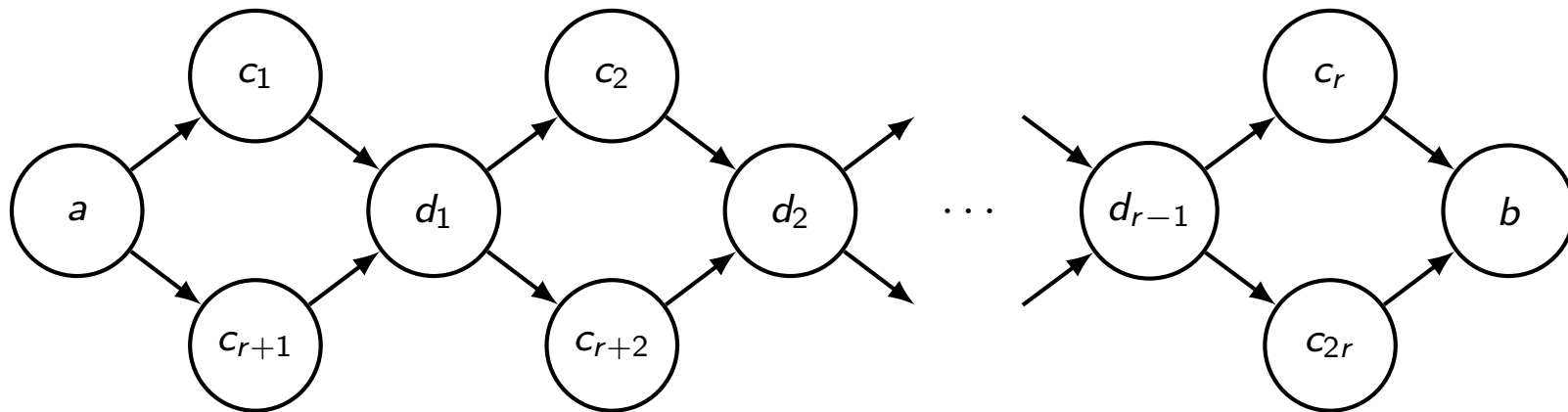


¿Cómo construimos el grafo aumentado?

Segundo intento: reemplazamos cada arco



por



donde $c_1, \dots, c_r, c_{r+1}, \dots, c_{2r}, d_1, \dots, d_{r-1}$ son nodos nuevos creados para reemplazar un arco.

El grafo aumentado

Sea G^r el grafo definido reemplazando cada arco como fue mostrado en la transparencia anterior.

Ejercicio

Defina formalmente el grafo G^r

Dado que G no tiene loops puede tener a lo más $n \cdot (n - 1)$ arcos

- ▶ Por lo tanto, G^r tiene a lo más $n + n \cdot (n - 1) \cdot (3 \cdot r - 1)$ nodos, y es de tamaño polinomial en el tamaño de G

La noción de testigo

Cada ciclo simple de G de largo $\ell \geq 2$ corresponde con $2^{\ell \cdot r}$ ciclos de G^r de largo $2 \cdot \ell \cdot r$

- Cada uno de estos ciclos de largo $2 \cdot \ell \cdot r$ en G^r es considerado como un testigo del ciclo de largo ℓ en G

Cada ciclo simple de G^r es testigo de un único ciclo simple de G

Ejercicio

Formalice la noción de testigo.

La noción de testigo

La conexión fundamental entre los grafos:

$G \in \text{HAM}$ si y sólo si G^r tiene un ciclo simple de largo $2 \cdot r \cdot n$

Utilizando esta conexión y eligiendo un valor adecuado para r vamos a construir un algoritmo aleatorizado de tiempo polinomial para HAM

- Teniendo como hipótesis la existencia de un FPRAS para $\#CicloSimple$

El número de testigos

Definimos los conjuntos:

$$\begin{aligned} I^r(G) &= \{C \mid C \text{ es un ciclo simple en } G^r \text{ de largo } 2 \cdot r \cdot n\} \\ M^r(G) &= \{C \mid C \text{ es un ciclo simple en } G^r \text{ de largo menor que } 2 \cdot r \cdot n\} \end{aligned}$$

Tenemos que $G \in \text{HAM}$ si y sólo si $I^r(G) \neq \emptyset$

- Suponiendo la existencia de un FPRAS para $\#\text{CicloSimple}$, vamos a desarrollar un algoritmo aleatorizado de tiempo polinomial para verificar si $I^r(G) \neq \emptyset$

El número de testigos

Para $\ell \in \{2, 3, \dots, n\}$ definimos t_ℓ como la cantidad de testigos en G^r para un ciclo simple en G de largo ℓ

Por la definición de G^r tenemos que $t_\ell = 2^{\ell \cdot r}$

Además, el número de ciclos simples en G de largo ℓ está acotado por:

$$\binom{n \cdot (n-1)}{\ell}$$

El número de testigos

A partir de lo anterior concluimos que:

$$\begin{aligned} |M^r(G)| &\leq \sum_{\ell=2}^{n-1} \binom{n \cdot (n-1)}{\ell} t_{\ell} \\ &\leq \sum_{\ell=2}^{n-1} \binom{n \cdot (n-1)}{\ell} t_{n-1} \\ &\leq t_{n-1} \sum_{\ell=0}^{n \cdot (n-1)} \binom{n \cdot (n-1)}{\ell} \\ &= 2^{n \cdot (n-1)} \cdot t_{n-1} \\ &= 2^{n \cdot (n-1)} \cdot 2^{(n-1) \cdot r} \\ &= 2^{(n-1) \cdot (n+r)} \end{aligned}$$

Idea fundamental: aumentando las diferencias

Por otra parte, si $G \in \text{HAM}$ tenemos que:

$$|I^r(G)| \geq t_n = 2^{n \cdot r}$$

Concluimos que:

- ▶ Si $G \notin \text{HAM}$: $\#\text{CicloSimple}(G^r) = |M^r(G)| \leq 2^{(n-1) \cdot (n+r)}$
- ▶ Si $G \in \text{HAM}$: $\#\text{CicloSimple}(G^r) \geq |I^r(G)| \geq 2^{n \cdot r}$

En las siguientes transparencias vamos a obtener un valor de r que nos permita tener un algoritmo aleatorizado de tiempo polinomial para HAM

Un algoritmo aleatorizado para HAM

Suponemos que existe un FPRAS $\mathcal{A}$ para $\#CicloSimple$, y mostramos como esto puede ser utilizado para obtener que $HAM \in BPP$

Definimos el siguiente algoritmo aleatorizado $\mathcal{B}$ para HAM:

1. Dada la entrada G , donde G es un grafo con n nodos, si $n \leq 1$ retorne **no**, en otro caso vaya al paso 2
2. Genere el grafo G^r
3. Sea s el resultado de ejecutar $\mathcal{A}(G^r, \frac{1}{2})$
4. Si $s > (1 + \frac{1}{2}) \cdot 2^{(n-1) \cdot (n+r)}$, entonces retorne **sí**, en caso contrario retorne **no**

Nótese que $\mathcal{B}$ funciona en tiempo polinomial en el tamaño de la entrada G si el valor de r es polinomial en n

- Recuerde que tenemos que determinar el valor de r

La probabilidad de error del algoritmo

Si $G \notin \text{HAM}$ obtenemos que $\Pr(\mathcal{B}(G) \text{ retorne sí})$ es igual a:

$$\begin{aligned} & \Pr(\mathcal{A}(G^r, \frac{1}{2}) > (1 + \frac{1}{2}) \cdot 2^{(n-1) \cdot (n+r)}) \\ & \leq \Pr(\mathcal{A}(G^r, \frac{1}{2}) > (1 + \frac{1}{2}) \cdot \#\text{CicloSimple}(G^r)) \\ & \leq \Pr(\mathcal{A}(G^r, \frac{1}{2}) > (1 + \frac{1}{2}) \cdot \#\text{CicloSimple}(G^r) \vee \\ & \quad \mathcal{A}(G^r, \frac{1}{2}) < (1 - \frac{1}{2}) \cdot \#\text{CicloSimple}(G^r)) \\ & = 1 - \Pr(\mathcal{A}(G^r, \frac{1}{2}) \leq (1 + \frac{1}{2}) \cdot \#\text{CicloSimple}(G^r) \wedge \\ & \quad \mathcal{A}(G^r, \frac{1}{2}) \geq (1 - \frac{1}{2}) \cdot \#\text{CicloSimple}(G^r)) \\ & = 1 - \Pr(|\mathcal{A}(G^r, \frac{1}{2}) - \#\text{CicloSimple}(G^r)| \leq \frac{1}{2} \cdot \#\text{CicloSimple}(G^r)) \\ & \leq 1 - \frac{3}{4} = \frac{1}{4} \end{aligned}$$

La probabilidad de error del algoritmo

Consideramos ahora el caso en que $G \in \text{HAM}$

Primero debemos encontrar un valor de r tal que:

$$\left(1 - \frac{1}{2}\right) \cdot \#\text{CicloSimple}(G^r) > \left(1 + \frac{1}{2}\right) \cdot 2^{(n-1) \cdot (n+r)}$$

Dado que $\#\text{CicloSimple}(G^r) \geq 2^{n \cdot r}$, basta entonces encontrar un valor de r tal que:

$$\left(1 - \frac{1}{2}\right) \cdot 2^{n \cdot r} > \left(1 + \frac{1}{2}\right) \cdot 2^{(n-1) \cdot (n+r)}$$

La probabilidad de error del algoritmo

Tenemos que:

$$\begin{aligned} & \left(1 - \frac{1}{2}\right) \cdot 2^{n \cdot r} > \left(1 + \frac{1}{2}\right) \cdot 2^{(n-1) \cdot (n+r)} \\ \Leftrightarrow & \frac{1}{2} \cdot 2^{n \cdot r} > \frac{3}{2} \cdot 2^{(n-1) \cdot (n+r)} \\ \Leftrightarrow & 2^{n \cdot r} > 3 \cdot 2^{(n-1) \cdot (n+r)} \\ \Leftrightarrow & 2^{n \cdot r} > 2^{(n-1) \cdot (n+r) + \log_2(3)} \\ \Leftrightarrow & n \cdot r > n \cdot (n-1) + (n-1) \cdot r + \log_2(3) \\ \Leftrightarrow & r > n \cdot (n-1) + \log_2(3) \end{aligned}$$

Considerando $r = n \cdot (n-1) + 2$ se cumple la condición.

► Concluimos que $\left(1 - \frac{1}{2}\right) \cdot \#\text{CicloSimple}(G^r) > \left(1 + \frac{1}{2}\right) \cdot 2^{n \cdot (n-1) + (n-1) \cdot r}$

La probabilidad de error del algoritmo

Si $G \in \text{HAM}$ y $r = n \cdot (n - 1) + 2$ tenemos que $\Pr(\mathcal{B}(G) \text{ retorne no})$ es igual a:

$$\begin{aligned} \Pr(\mathcal{A}(G^r, \frac{1}{2}) \leq (1 + \frac{1}{2}) \cdot 2^{(n-1) \cdot (n+r)}) \\ &\leq \Pr(\mathcal{A}(G^r, \frac{1}{2}) < (1 - \frac{1}{2}) \cdot \#\text{CicloSimple}(G^r)) \\ &\leq \Pr(\mathcal{A}(G^r, \frac{1}{2}) < (1 - \frac{1}{2}) \cdot \#\text{CicloSimple}(G^r) \vee \\ &\quad \mathcal{A}(G^r, \frac{1}{2}) > (1 + \frac{1}{2}) \cdot \#\text{CicloSimple}(G^r)) \\ &= 1 - \Pr(\mathcal{A}(G^r, \frac{1}{2}) \geq (1 - \frac{1}{2}) \cdot \#\text{CicloSimple}(G^r) \wedge \\ &\quad \mathcal{A}(G^r, \frac{1}{2}) \leq (1 + \frac{1}{2}) \cdot \#\text{CicloSimple}(G^r)) \\ &= 1 - \Pr(|\mathcal{A}(G^r, \frac{1}{2}) - \#\text{CicloSimple}(G^r)| \leq \frac{1}{2} \cdot \#\text{CicloSimple}(G^r)) \\ &\leq 1 - \frac{3}{4} = \frac{1}{4} \end{aligned}$$

#CicloSimple no admite un FPRAS: conclusión

Tenemos que en ambos casos el error del algoritmo $\mathcal{B}$ está acotado superiormente por $\frac{1}{4}$

- ▶ Considerando $r = n \cdot (n - 1) + 2$ en ambos casos

Concluimos entonces que $\text{HAM} \in \text{BPP}$

- ▶ Por lo tanto $\text{NP} \subseteq \text{BPP}$



La existencia de FPRAS y las reducciones parsimoniosas

Sean $f, g : \Sigma^* \rightarrow \mathbb{N}$ dos funciones en $\#P$

Teorema

Si $f \leq_{par}^p g$ y existe un FPRAS para g , entonces existe un FPRAS para f

Demostración: Suponga que $h : \Sigma^* \rightarrow \Sigma^*$ es una función computable en tiempo polinomial tal que $f(w) = g(h(w))$ para todo $w \in \Sigma^*$

La existencia de FPRAS y las reducciones parsimoniosas

Además, suponga que $\mathcal{A}$ es un FPRAS para g , y defina un algoritmo aleatorizado $\mathcal{B} : \Sigma^* \times \mathbb{R}^+ \rightarrow \mathbb{N}$ de la siguiente forma:

Para cada $w \in \Sigma^*$ y $\varepsilon \in \mathbb{R}^+$: $\mathcal{B}(w, \varepsilon) = \mathcal{A}(h(w), \varepsilon)$

Para cada $w \in \Sigma^*$ y $\varepsilon \in \mathbb{R}^+$:

$$\begin{aligned} \Pr\left(|\mathcal{B}(w, \varepsilon) - f(w)| \leq \varepsilon \cdot f(w)\right) \\ &= \Pr\left(|\mathcal{A}(h(w), \varepsilon) - g(h(w))| \leq \varepsilon \cdot g(h(w))\right) \\ &\geq \frac{3}{4} \end{aligned}$$

Por lo tanto $\mathcal{B}$ es un FPRAS para f

□

Utilizando las reducciones parsimoniosas

Suponiendo que $f \leq_{par}^p g$, el resultado anterior puede utilizarse de dos formas:

- ▶ Si tenemos un FPRAS para g , entonces podemos construir un FPRAS para f
- ▶ Si tenemos una demostración que no existe un FPRAS para f , entonces concluimos que no existe un FPRAS para g

Vamos a encontrar otras funciones que no admiten FPRAS utilizando el resultado anterior.

Contando el número de cliques

Dado un grafo $G = (N, A)$, decimos que $S \subseteq N$ es un clique si para cada $a, b \in S$ se tiene que $(a, b) \in A$

Sea $\#CLIQUE$ una función que, dado un grafo G , retorna el número de cliques de G

Teorema

Si existe un FPRAS para $\#CLIQUE$, entonces $NP \subseteq BPP$

No existe un FPRAS para $\# \text{CLIQUE}$: demostración

Dado un grafo $G = (N, A)$, defina $\overline{G} = (N, \overline{A})$ con $\overline{A} = (N \times N) \setminus A$

Tenemos que $\# \text{IS}(G) = \# \text{CLIQUE}(\overline{G})$

Concluimos que $\# \text{IS} \leq_{\text{par}}^P \# \text{CLIQUE}$

- Por lo tanto, si existe un FPRAS para $\# \text{CLIQUE}$, entonces existe un FPRAS para $\# \text{IS}$ y $\text{NP} \subseteq \text{BPP}$ □

Otro ejemplo: las cláusulas de Horn

Recuerde que una cláusula se dice de Horn si contiene a lo más un literal positivo.

Ejemplo

$(p \vee \neg q \vee \neg r)$, p , $(\neg q \vee \neg r)$ y $\neg q$ son cláusulas de Horn, mientras que $(p \vee q)$ y $(p \vee \neg q \vee r \vee \neg s)$ no lo son.

Además, decimos que una fórmula proposicional φ es de Horn si φ es una conjunción de cláusulas de Horn.

La función $\#HORN-SAT$

Sea $\#HORN-SAT$ una función que, dada una fórmula proposicional φ de Horn, retorna el número de valuaciones que satisfacen φ

Teorema

$\#HORN-SAT$ es $\#P$ -completo.

Ejercicio

Demuestre que $L_{\#HORN-SAT} \in PTIME$

No existe un FPRAS para #HORN-SAT

Teorema

Si existe un FPRAS para #HORN-SAT, entonces $NP \subseteq BPP$

Demostración: Sea $G = (N, A)$ un grafo tal que $N \neq \emptyset$

Por cada $v \in N$, considere una variable proposicional x_v , y defina φ_G como la siguiente fórmula proposicional:

$$\left(\bigwedge_{a \in N} (x_a \vee \neg x_a) \right) \wedge \left(\bigwedge_{(b,c) \in A} (\neg x_b \vee \neg x_c) \right)$$

No existe un FPRAS para $\# \text{HORN-SAT}$: demostración

Tenemos que φ_G es una fórmula proposicional de Horn
y $\# \text{IS}(G) = \# \text{HORN-SAT}(\varphi_G)$

- ▶ ¿Cómo se puede extraer un conjunto independiente de G desde una valuación que satisface φ_G ?

Concluimos que $\# \text{IS} \leq_{\text{par}}^p \# \text{HORN-SAT}$

- ▶ ¿Cómo se maneja el caso en que $N = \emptyset$ en la reducción?

Por lo tanto, si existe un FPRAS para $\# \text{HORN-SAT}$, entonces existe un FPRAS para $\# \text{IS}$ y $\text{NP} \subseteq \text{BPP}$ □