Práctica 6:

Ej 1:

1. Probar que el lenguaje Σ_iSAT es completo para la clase Σ_i^p, donde

   - Σ_i SAT $= \{\langle \phi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i)\rangle : \phi$ es una fórmula booleana y $\exists \mathbf{v}_1 \forall \mathbf{v}_2 \ldots \phi(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_i)\}$

No lo hice, ¿mi justificación? Está en la teórica (Corolario 6 de la prop. 32)

Idea: Muy similar al Teorema de Cook-Levin con mini configuraciones y todo eso...

Ej 2:

2. Probar que si $\Sigma_k^p = \Pi_k^p$ para algún $k \in \mathbb{N}$, entonces $\mathsf{PH} = \Sigma_k^p$.

Es similar a la prop. 28 (si $P = NP \Rightarrow PH = P$).

Si $\Sigma_k^p = \Pi_k^p$ para algún $k \in \mathbb{N}$, entonces voy a probar q' para cualquier Σ_i^p, Π_i^p con $i \geq k$, estos son iguales a Σ_k^p y Π_k^p.

Inducción:

- Caso base: Con Σ_k^p y Π_k^p es trivial
- Caso inductivo:

HI: Vale para Σ_i^p y Π_i^p.

Luego, sea $L \in \Sigma_{i+1}^p$. Por def., existe una máq. det. poly. M y un polinomio p tal que:

$$x \in L \text{ sii } \exists u_1 \forall u_2 \ldots Q_{i+1} u_{i+1} .\ M(x, u_1, u_2, \ldots, u_{i+1}) = 1 \quad (\text{con } u_1, \ldots, u_{i+1} \in \{0,1\}^{p(|x|)})$$

Defino L' tal que:

$$\langle x, u_1 \rangle \in L' \text{ sii } \underbrace{\forall u_2 \ldots Q_{i+1} u_{i+1} .\ M(\langle x, u_1, u_2, \ldots, u_{i+1}\rangle) = 1}^{\text{Esta es mi HI (tiene } i \text{ cuantificadores y arranca con } \forall \text{, es de } \Pi_i^p)}$$

$(\text{con } u_1, \ldots, u_{i+1} \in \{0,1\}^{p(|x|)})$

Por HI:

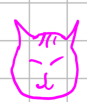
$$\langle x, u_1 \rangle \in L' \text{ sii } \exists u_2 \ldots Q_k u_k\ M(x, u_1, u_2, \ldots, u_k) = 1 \quad (u_1, \ldots, u_k \in \{0,1\}^{p(|x|)})$$

Entonces, para todo $x \in \{0,1\}^*$:

$$x \in L \text{ sii } \exists u_1 u_2 \ldots Q_k u_k\ M(x, u_1, u_2, \ldots, u_k) = 1 \quad (u_1, \ldots, u_k \in \{0,1\}^{p(|x|)}).$$

Lo cual es Σ_k^p

★ Es casi idéntico para Π_{i+1}^p (no lo pienso hacer, pero imaginate que está hecho)



Gatito para descontracturar

Ej 3:

3. Probar que si SAT $\leq_p \overline{\text{SAT}}$ entonces $\mathsf{PH} = \mathsf{NP}$.

$x \in$ SAT sii $f(x) \in \overline{\text{SAT}}$ (con f computable en tiempo poly)

Si puedo reducir un problema NP-Completo a uno coNP, entonces puedo reducir cualquier problema de NP a coNP. ($NP \subseteq coNP$)
A su vez, como ahora sé q' SAT $\in$ coNP puedo decir q' su complemento ($\overline{SAT}$) debe estar en NP (por déf. de coNP), esto implica q' tengo un problema coNP-completo en NP, o sea q' puedo reducir cualquier $\pi \in$ coNP a uno NP. ($coNP \subseteq NP$).

Con esto llego a q' NP = coNP, o sea $\Sigma_1^P = \Pi_1^P$ y a partir de acá es la misma demo q' en el punto anterior.

Ej 4:

4. Probar que el problema FORMULA_MAS_CHICA de la guía anterior está en Π_2^p.

FORMULA_MAS_CHICA: $\{\langle \phi, K\rangle : \phi$ es una fórmula booleana proposicional, y no existe fórmula ϕ' tal q' $\phi \equiv \phi'$ y $|\phi'| \leq K\}$.

Idea: Hacer cadena de implicaciones hasta llegar a la definición de Π_2^P.

$$\underbrace{}^{*'}$$

Recibo $\langle \phi, K\rangle$, ϕ es una fórmula booleana proposicional, y no existe una fórmula ϕ' tal que $\phi \equiv \phi'$ y $|\phi'| \leq K$. sii

(*' Siempre está, no lo pongo para no tener tantas repeticiones)

$\neg\exists \phi'.\ \phi \equiv \phi'$ y $|\phi'| \leq K$ sii

$\neg\exists \phi'.\ \forall v\ (v \models \phi \Leftrightarrow v \models \phi')$ y $|\phi'| \leq K$ sii

$\forall \phi' \exists v.\ \neg((v \models \phi \Leftrightarrow v \models \phi')$ y $|\phi'| \leq K)$ sii

$\forall \phi' \exists v.\ |\phi'| > K$ ó $\neg((v \models \phi \Rightarrow v \models \phi')$ y $(v \models \phi' \Rightarrow v \models \phi))$ sii

$\forall \phi' \exists v.\ |\phi'| > K$ ó $\neg(v \models \phi \Rightarrow v \models \phi')$ ó $\neg(v \models \phi' \Rightarrow v \models \phi)$ sii

$\forall \phi' \exists v.\ |\phi'| > K$ ó $\neg(v \not\models \phi$ ó $v \models \phi')$ ó $\neg(v \not\models \phi'$ ó $v \models \phi)$ sii

$$\forall \phi' \exists v.\ \underbrace{|\phi'| > K \text{ ó } (v \models \phi \text{ y } v \not\models \phi') \text{ ó } (v \models \phi' \text{ y } v \not\models \phi)}_{\text{Se puede hacer con una máq. det. poly.}}$$

Entonces queda q':

$\forall \phi' \exists v.\ M(\langle\phi, K\rangle, \phi', v\rangle)$ con M det. y poly, lo cual es la definición de Π_2^P.

Ej 5:

5. La clase DP $= \{L_1 \cap L_2 : L_1 \in \mathsf{NP}, L_2 \in \mathsf{coNP}\}$ consiste de la intersección de problemas que están en NP y coNP (notar que $\mathsf{DP} \neq \mathsf{NP} \cap \mathsf{coNP}$). Probar que:

   a) $\mathsf{DP} \subseteq \Sigma_2^p \cap \Pi_2^p$.

   b) El siguiente lenguaje está en DP.

      - EXACT INDSET $= \{\langle G, k\rangle : G$ es un grafo cuyo conjunto independiente más grande tiene tamaño $k\}$

   c) EXACT INDSET es completo para DP. Para esto, seguir la siguiente estrategia:

      1) Dar una reducción g de SAT a INDSET (el problema de dado un grafo G y un k decidir si G tiene un conjunto independiente de tamaño mayor o igual a k). Basar la misma en la siguiente idea: por cada cláusula definir una clique de tamaño 7 que represente las 7 asignaciones que satisfacen la cláusula. Luego, conectar todos los nodos que representan asignaciones inconsistentes.

      2) Observar que la reducción propuesta en el ejercicio anterior puede modificarse de tal forma que si $\varphi(x_1, \ldots, x_n)$ es satisfacible entonces el conjunto independiente más grande de G tiene tamaño n, mientras que si φ no lo es entonces el conjunto independiente más grande tiene tamaño $n-1$.

      3) Dado un lenguaje $\Pi \in \mathsf{DP}$ con $L = \Pi_1 \cap \Pi_2, \Pi_1 \in \mathsf{NP}, \Pi_2 \in \mathsf{coNP}$, sean las reducciones f_1 y f_2 de Π_1 a SAT y de Π_2 a $\overline{\text{SAT}}$. Dado x, considerar las reducciones $g(f_1(x)) = \langle G_1, k_1\rangle$ y $g(f_2(x)) = \langle G_2, k_2\rangle$. Suponiendo que $k_1 \neq k_2$, probar que $x \in \Pi$ si y solamente si $\langle G_1 \times G_2, k_1(k_2-1)\rangle \in$ EXACT INDSET [1].

      4) Adaptar el argumento para el caso en que $k_1 = k_2$.

a) $DP \subseteq \Sigma_2^p \cap \Pi_2^p = NP^{NP} \cap coNP^{NP}$

Si $\mathcal{L} \in DP$, entonces: $\mathcal{L} = \mathcal{L}_1 \cap \mathcal{L}_2$ con $\mathcal{L}_1 \in NP$ y $\mathcal{L}_2 \in coNP$

Por def. si $\mathcal{L}_1 \in NP$: (M det. poly y p es una función polinomial).

$x \in \mathcal{L}_1$ sii $\exists u_1 \in \{0,1\}^{p(|x|)} . M(x, u_1) = 1$

Por def. si $\mathcal{L}_2 \in coNP$, $\overline{\mathcal{L}_2} \in coNP$ tal q': (M' det. poly, p función polinomial)

$x \in \mathcal{L}_2$ sii $x \notin \overline{\mathcal{L}_2} \Rightarrow x \in \mathcal{L}_2$ sii $\forall u_2 \in \{0,1\}^{p(|x|)}\ M'(\langle x, u_2\rangle) = 0$.

M' es el verificador det. y poly de $\mathcal{L}_2$.

Luego, puedo decir q':

$x \in \mathcal{L}$ sii $x \in \mathcal{L}_1 \cap \mathcal{L}_2$ sii $\underbrace{\exists u_1 \forall u_2 . M(x, u_1) = 1 \text{ y } M'(x, u_2) = 0}_{\in \Sigma_2^p}$

Así queda probado q' $\mathcal{L} \in \Sigma_2^p$.

Para probar q' $\mathcal{L} \in \Pi_2^p$ es similar:

$x \in \mathcal{L}$ sii $x \in \mathcal{L}_1 \cap \mathcal{L}_2$ sii $x \notin \overline{\mathcal{L}_1} \cup \overline{\mathcal{L}_2}$ $\quad \overline{(\mathcal{L}_1 \cap \mathcal{L}_2)} \equiv \overline{\mathcal{L}_1} \cup \overline{\mathcal{L}_2}$

Después veo q $\mathcal{L} \in \Pi_2^p$ viendo a $\mathcal{L} = \overline{\overline{\mathcal{L}_1} \cup \overline{\mathcal{L}_2}}$:

$x \in \mathcal{L}_1$ sii $x \notin \overline{\mathcal{L}_1} \Rightarrow x \in \overline{\mathcal{L}_1}$ sii $\forall u_1\ M(x, u_1) = 0$

con M verificador de $\mathcal{L}_1$ ya q' $\mathcal{L}_1 \in NP$.

$x \in \mathcal{L}_2$ sii $x \notin \overline{\mathcal{L}_2} \Rightarrow x \in \overline{\mathcal{L}_2}$ sii $\exists u_2 . M'(x, u_2) = 1$

con M' verificador de $\overline{\mathcal{L}_2}$ pq' $\mathcal{L}_2 \in coNP$.

Después, $x \in \mathcal{L}$ sii $x \in \overline{\overline{L_1} \cup \overline{L_2}}$; $x \notin \mathcal{L}$ sii $x \in \overline{L}$

$x \in \mathcal{L}$ sii $\underbrace{\forall u_2 \exists u_1 . M(x,u_1) = 1 \text{ ó } M'(x,u_2) = 0}_{*^2 \in \Pi_2^p}$

γ (lo último justificación tomarla con pinzas)

b)

$\mathcal{L} = \mathcal{L}_1 \cap \mathcal{L}_2$ con $\mathcal{L}_1 \in NP$ y $\mathcal{L}_2 \in coNP$

$x \in \mathcal{L}$ sii $\underbrace{\text{G tiene un cjto de tamaño K}}_{\text{con INDSET\_EQUAL} = \mathcal{L}_1}$ y $\underbrace{\text{no tiene uno de tamaño mayor}}_{\text{coINDSET\_EQUAL} = \mathcal{L}_2}$

$\mathcal{L}_1 \in NP$:

$M(\langle G,K\rangle, c)$: $\overset{\text{verificador}}{M}$ es det. poly y $c \in \{0,1\}^{p(|x|)}$ ($x \mapsto \langle G,K\rangle$) con C = cjto. de nodos de tamaño K.

Algoritmo:

Chequeo q' c sea un cjto independiente de G.

↳ Que cada nodo no está conectado con ningún otro del cjto.

$\mathcal{L}_2 \in coNP$:

$M'(\langle G,K\rangle, c)$: M es det. poly y $c \in \{0,1\}^{p(|x|)}$ con C = Cjto de nodos con tamaño j con $K < j \leq |V|$.

Algoritmo: (de $\overline{L_2}$ q' $\in NP$, $\overline{L_2}$ = hay un cjto independiente de tamaño mayor a K)

Chequeo q' el c sea un cjto independiente de G de tamaño mayor a K.

γ

c) Muy complicado, algo así no va a ser evaluado en el exámen pero me gustaría hacerlo eventualmente

γ