Práctica 7:

Ej 1:

1. Probar que para todo oráculo $\mathcal{A}$ se tiene $\mathsf{P}^{\mathcal{A}} \subseteq \mathsf{NP}^{\mathcal{A}} \subseteq \mathsf{E}^{\mathcal{A}}$.

Sé q' $L \in P^A$ implica q' hay una máq. det. (M) q' usa el oráculo A.

Luego, puedo armar N (det.) q' tenga de certificado ε (vacío) y q' su funcionamiento sea igual al del de M. Entonces N sería el verificador y prueba q' $L \in NP^A$ (como es tan general el L me permite representar todo P).

Por último, hago una Z (det.) q' genere todos los certificados posibles de N ($2^{p(|x|)}$ con $p(|x|)$ el tamaño del certificado) y luego imito el comportamiento de N para cada uno de ellos (las consultas al oráculo son iguales).

Ej 2:

2. Considerar el siguiente prolema:

   - LEX-SAT-BIT $= \{\langle \varphi, i \rangle : \varphi$ es una fórmula satisfacible y la menor asignación que la satisface (donde menor se define usando el orden lexicográfico) fija la variable i en 1$\}$

   Probar LEX-SAT-BIT $\in \mathsf{P}^{\mathsf{NP}}$. Argumentar por qué el problema no debería estar en NP.

Este ejercicio fue hecho en la práctica (no me lo acuerdo igual :/)

El algoritmo consiste en ir lexicográficamente de menor a mayor armando la valuación de φ más chica.

Por cada iteración analizo si le pongo un 0, ¿sigue siendo SAT?, si la rta es que no, entonces le pongo un 1.

Arranco desde el bit más significativo preguntando hacia el menor así de esta manera consigo la menor lexicográficamente.

Una vez ya obtenida la menor valuación de φ retorno si $V[i] == 1$.

La intuición de por qué no debería estar en NP, es que tengo que chequear q' la valuación que tengo sea la menor y para esto debo ver que ninguna menor es satisfacible, Compliqueti no hacerlo iterando por todas y viendo que cada una no sea sat.

Ej 3:

3. Probar que $\mathsf{NP} \cup \mathsf{coNP} \subseteq \mathsf{P}^{\mathsf{NP}}$.

QVQ si $L \in NP \cup coNP \Rightarrow L \in P^{NP}$.

Si $L \in NP$:

Puedo usar q' el oráculo de la máq. det. poly. sea L y lo único q' hago la máquina es llamar al oráculo.

Si $L \in coNP$

Tomo $\overline{L}$ como oráculo, le paso la entrada al oráculo y retorno la negación de la respuesta del oráculo

γ

Ej 4:

4. Probar que $\mathsf{P}^{\mathsf{NP}} \subseteq \Sigma_2^p \cap \Pi_2^p$. Generalizar a $\mathsf{P}^{\Sigma_i^p} \subseteq \Sigma_{i+1}^p \cap \Pi_{i+1}^p$ para todo $i \in \mathbb{N}$.

Si $L \in P^{NP} \Rightarrow L \in \Sigma_2^P \cap \Pi_2^P$

Luego, L entonces es decidido por una máq. M det. poly. con oráculo (a lo sumo NP-completo)

Q.V.Q $L \in \Sigma_2^P$ y $L \in \Pi_2^P$ q' es equivalente (por lo visto en lo teórico) a que $L \in NP^{NP}$ y $L \in coNP^{NP}$.

Si $L \in P^{NP} \Rightarrow L \in NP^{NP}$:

Creo una máq. no-det la cuál usa el oráculo NP-completo, reduzco el oráculo de M a este nuevo y hago una máq no-det q' haga lo mismo q' M.

Si $L \in P^{NP} \Rightarrow L \in coNP^{NP}$:

De L puedo decir o sea q' $\bar{L} \in P^{NP}$, ya q' P^{NP} está cerrada por complemento, $\bar{L} \in coP^{NP} = P^{NP}$ q' $\Rightarrow \bar{L} \in NP^{NP}$ (por lo del anterior párrafo) $\Rightarrow L \in coNP^{NP}$.

- La generalización es posible con inducción del Teo 26 q' declara q' $\Sigma_{i+1}^P = NP^{\Sigma_i SAT}$. y

Ej 5:

5. Probar que $\mathsf{NP}^{\mathsf{NP}} = \Sigma_2^p$. Generalizar esto para caracterizar todos los pisos de la jerarquía polinomial en función de oráculos.

Es la demo del Teo 26.

Ej 6:

6. Probar que existen lenguajes $\mathcal{A}, \mathcal{B} \in \mathsf{EXP}$ tales que $\mathsf{P}^{\mathcal{A}} = \mathsf{NP}^{\mathcal{A}}$ y $\mathsf{P}^{\mathcal{B}} \subsetneq \mathsf{NP}^{\mathcal{B}}$. **Ayuda**: Para $\mathcal{B}$, considerar la siguiente idea: proponer un lenguaje $\mathcal{B}$ tal que el problema

$$U_B = \{1^n : \exists x \in \{0,1\}^n \text{ tal que } x \in B\} \tag{1}$$

no se pueda resolver en $\mathsf{P}^{\mathcal{B}}$. Armar $\mathcal{B}$ diagonalizando todas las máquinas polinomiales.

La resolución sería un caso específico del Teo 25 de Baker, Gill y Solovay. (Igual esto es la demo idéntica).

Ej 7:

7. Probar que existen lenguajes $\mathcal{A}, \mathcal{B}$ tales que $\mathsf{NP}^{\mathcal{A}} = \mathsf{coNP}^{\mathcal{A}}$ y $\mathsf{NP}^{\mathcal{B}} \neq \mathsf{coNP}^{\mathcal{B}}$. **Ayuda**: usar las ideas del ejercicio anterior.

Similar al anterior.

Ej 8:

8. Probar que $\mathsf{PSPACE}^{\mathsf{PSPACE}} = \mathsf{PSPACE}$.

1) $PSPACE \subseteq PSPACE^{PSPACE}$ → Trivial ^^

2) $PSPACE \subseteq PSPACE^{PSPACE}$

Creo una máq. M det. poly con un oráculo PSPACE-completo (TQBF por ejemplo)

$L \in PSPACE^{TQBF}$ → Decidible con M det. q' corre en espacio poly y tiene de oráculo a TQBF.

Luego, puedo reemplazar M con una máq. det. q' por cada llamado al oráculo simula una máq. det. q' decide TQBF en PSPACE en el tamaño de entrada de la query.

El espacio q' ocupa la query del oráculo es poly respecto a la entrada y el espacio de ejecución de la máq. q' decide TQBF es poly respecto a esta, o sea es poly(poly), por lo cual es poly respecto a la entrada en general. ✓

Ej 9:

9. Encontrar una clase C tal que $C^C \neq C$.

Puedo tomar $E \neq E^E$. Sé q' $E \subseteq E^E$, pero ¿$E^E \subseteq E$?

La respuesta corta es no... ¿Por qué?

Si yo tengo una M det. con un oráculo $\pi \in E^E$-completo (M' decide π) q' decide un $L \in E^E$-completo, no puedo simular el oráculo como antes porque para cada una de las querys hechas al oráculo exponencial respecto a la entrada de M vuelve la ejecución de M' exponencial respecto al tamaño de la query, o sea 2^E respecto a la entrada de la máquina (2^{2^n}).

Por jerarquía temporal, sé q' $E \subsetneq 2^E$. ✓

Ej 10:

10. Probar que $\mathsf{NP}^{\mathsf{NP} \cap \mathsf{coNP}} = \mathsf{NP}$.

El = es una doble inclusión, así q' como dijo Jack el destripador, vamos por partes...

$\supseteq$) $NP \subseteq NP^{NP \cap coNP}$ → Es trivial, el oráculo no restringe el hecho de hacer lo mismo.

$\subseteq$) $NP^{NP \cap coNP} \subseteq NP$

$\exists N$. N no-det y N decide L con $L \in NP^{NP \cap coNP}$, N hace llamados al oráculo A ($A \in NP \cap coNP$).

Para simular esto con M no-det. poly. tengo q' ver qué hacer.

Como $A \in NP \cap coNP$, $x \in A$ ó $x \notin A$ son cosas verificables en tiempo polinomial.

Por cada llamada al oráculo entonces podría reemplazarla por una verificación no-det. de la entrada con un certificado inventado (como se hace en NP).

Entonces, se concluye q' como puedo simular la rta del oráculo si este está en $NP \cap coNP$, no tiene más poder por el agregado de este, una máq. no-det. poly.

Ej 11:

11. Dada una clase $\mathcal{C}$, se define $\mathtt{low}(\mathcal{C}) = \{\Pi \subseteq \Sigma^\star : \mathcal{C}^\Pi = \mathcal{C}\}$. Probar que $\mathtt{low}(\mathsf{NP}) = \mathsf{NP} \cap \mathsf{coNP}$.

low(C) = Significa q' el oráculo π no le agrega poder a un problema q' $\in NP$.

Si usamos $\pi \in NP \cap coNP$ por el pto. 10 sabemos q' esto funcionaría.

↳ $NP \cap coNP \subseteq low(NP)$

Ahora, $low(NP) \subseteq NP \cap coNP$?

Si: $L \notin NP \cap coNP$:

- O bien $L \notin NP$ ó $L \notin coNP$ (o ambos)
- Entonces hay un L' (p_{ej}: $x \in L$?) q' no se puede decidir en NP pero sí en NP^L (con 1 consulta)
  - ↳ Contradice "$NP^L = NP$" pq' con el oráculo sí ganas poder.

Entonces, si: $L \notin NP \cap coNP \Rightarrow L \notin low(NP)$.

Por último, queda probado q':

$$low(NP) = NP \cap coNP$$

γ