Teoremas, prop, etc. (Mi punto de vista de demos)

obs: $L \subseteq NL \subseteq P$

Demo: Si S es construible en espacio, sabemos q' $NSpace(S(n)) \subseteq Space(2^{O(S(n))})$.
Tomar $S(n) = \log(n)$.

Proposición 23. PATH $\in$ **NL**.

Demo: Inventa un camino de s a t, almacenando 2 nodos únicamente ($O(2\log(n))$), la longitud del camino es de a lo sumo $n-1$ ($n = |V|$).

$y \leftarrow s$; $m \leftarrow 0$
mientras $m < |V|$:
  $z \leftarrow$ inventar un valor en $\{0, \ldots, |V|-1\}$
  si $(y, z) \notin E$ (para esto, revisa G en la entrada), pasar a q_{no}
  si no:
    si $z = t$, pasar a $q_{sí}$
    si no:
      $y \leftarrow z$; $m \leftarrow m + 1$
pasar a q_{no}

→ Es útil saberse el algoritmo concreto, pq' es usado algo similar en muchos casos.

Proposición 24. *Si $\mathcal{L} \notin \{\{0,1\}^*, \emptyset\}$ entonces $\mathcal{L}$ es* **NL-hard** *con respecto a* $\leq_p$.

Demo:

Puedo reducir a cualquier $\mathcal{L}' \in NL$ a un $\mathcal{L} \neq$ de los triviales.

Es absurdo pq' puedo computar $\chi_{\mathcal{L}'}$ en tiempo polinomial directamente, ya q' $NL \subseteq P$.

Proposición 25. *Sean f, g computables implícitamente en* **L**. *Entonces $g \circ f$ es computable implícitamente en* **L**.

Demo:

Sale de ir computando M_g y cada vez que requiera un bit de $f(x)$, lo computo con M_f.

$$\left.\begin{array}{l} M_g(\langle x, i\rangle) = g(x)(i) \\ M_f(\langle x, i\rangle) = f(x)(i) \end{array}\right\} \text{Bit } i \text{ de la salida de hacer } g/f \text{ con entrada } x.$$

Teorema 21. PATH $\in$ **NL-completo** *y por lo tanto* $\overline{\text{PATH}} \in$ **coNL-completo**.

Demo:

PATH $\in$ NL-HARD: Si $f(x) = \langle G_{N,x}, C_0, C_f\rangle$, con C_0 = Configuración inicial de N con entrada x, C_f = Configuración final.

Cada configuración se codifica con $c \cdot \log|x|$ bits.

Luego,

$x \in \mathcal{L}$ sii N acepta x

sii Existe un camino desde C_0 a C_f en $G_{N,x}$

sii $f(x) \in$ PATH

O sea, básicamente es una especie de Reach, pero implementada con PATH, donde no me guardo el camino q' hago.

Todavía no está todo, hay que ver que f es computable implícitamente en L:

Numeramos todas las configuraciones en orden lexicográfico.

$G_{N,x}$ es representable con una matriz de adyacencia de $(2^{c \cdot \log|x|})^2$ cant. de configuraciones

Luego, $|f(x)| = O((2^{c \cdot \log|x|})^2) = O(|x|^{2c})$

Uso el truquito de obtener la matriz de adyacencia on-the-fly.

O sea, tengo M det. tal q' dada $\langle x, i \rangle$ decide si i corresponde a un bit de la matriz de adyacencia de $G_{N,x}$, o a un bit de la codificación de C_0 o a un bit de C_f.

PATH $\in$ coNL-completo:

Si sé que para cualquier $\mathcal{L} \in NL$, $\mathcal{L} \leq_l$ PATH, puedo decir q' puedo ver que puedo, para cualquier $\overline{\mathcal{L}} \in coNL$ buscar $\mathcal{L} \in NL$ y reducirla a PATH, donde su complemento equivaldría a $\overline{\mathcal{L}}$.

$$\underbrace{x \in \overline{\mathcal{L}} \text{ sii } x \notin \mathcal{L} \text{ sii } f(x) \notin \text{PATH sii}}_{\text{Esto es computable implícitamente en } L \text{, por lo cual: }} f(x) \in \overline{\text{PATH}}$$

$\overline{\mathcal{L}} \leq_l \overline{\text{PATH}}$ ^^

Teorema 22. $\mathcal{L} \in \mathbf{NL}$ *sii existe un polinomio* $p : \mathbb{N} \to \mathbb{N}$ *y una máquina determinística* M *con una cinta de entrada adicional de lectura de una única vez (lee y pasa a la siguiente celda a la derecha pero no puede volver atrás) tal que 1) para todo* $x \in \{0,1\}^*$, $x \in \mathcal{L}$ *sii existe* $u \in \{0,1\}^{p(|x|)}$ *tal que* $M(x,u) = 1$*; aquí* $M(x,u)$ *denota la salida de* M *cuando la cinta de entrada tiene* x *y la cinta adicional de lectura de una única vez tiene* u*; 2)* M *usa espacio* $O(\log n)$*; en este modelo de máquina con cinta de entrada adicional, la cinta de entrada y la cinta adicional no cuentan en el espacio (solo cuentan sus cintas de trabajo y salida).*

Demo: Es lo análogo a lo de NP con certificados para NL.

Teorema de Immerman-Szelepcsényi

Teorema 23 (Immerman-Szelepcsényi). $\overline{\text{PATH}} \in \mathbf{NL}$.

Demo:

QvQ $\langle G, s, t \rangle \notin$ PATH se puede verificar en tiempo poly.

Sea $G = (V, E)$ y $V = \{1, \dots, n\}$. Sea el lenguaje:

$$A_i = \{ v \in V : v \text{ es alcanzable desde } s \text{ en a lo sumo } i \text{ pasos} \}$$

Galerazo

Ver q' A_n es la componente conexa de s en G (Todos los nodos a los q' puede llegar).

$$\langle G, s, t \rangle \notin \text{PATH} \quad \text{sii} \quad t \notin A_n.$$ (Es una reducción bastante fácil de ver)

Luego qvq la no pertenencia de t en A_n está en NL.

Ahora toca ver q' hay un certificado y verificador q' permite ver q' $t \notin A_n$

Partimos el certificado en distintos Z para distintos hechos.

Certificado de q' $v \in A_i$:

$$Z_{v \in A_i} = \langle v_0, v_1, \dots, v_k \rangle$$

$v_0 = s$; v_i $(0 \leq i \leq k)$ es la codificación de un nodo de V; $(v_i, v_{i+1}) \in E$;
$v_k = v$; $k \leq i$.

*' Notar q' $|Z_{v \in A_i}|$ es poly y q' el verificador puede chequear el certificado en $O(\log(n))$.

Certificado de q' $v \notin A_i$ conociendo $|A_i|$

Similar al anterior,

$$Z^{|A|}_{v \notin A_i} = \langle (v_1, Z_{v_1 \in A_i}), \overbrace{(v_2, Z_{v_2 \in A_i}), \dots}^{\text{Todos son vdd.}}, (v_k, \underbrace{Z_{v_k \in A_i}}_{}) \rangle$$

En A_i se alcanzan exáctamente $|A_i|$ nodos

$v_i \in V$; $\underline{k = |A_i|}$; $v_i < v_{i+1}$; $v \notin \{v_1, \dots, v_k\}$

$Z^{|A_i|}_{v \notin A_i}$ = Muestra k elementos distintos en A_i, tal q' ~~ninguno es v~~.

$|A_i|$ = Cantidad de nodos alcanzables desde s en i pasos *'

Certificado de q' $v \notin A_i$ conociendo $|A_{i-1}|$

Similar a lo anterior,

$$Z^{|A_{i-1}|}_{v \notin A_i} = \langle (v_1, Z_{v_1 \in A_{i-1}}), \dots, (v_k, Z_{v_k \in A_{i-1}}) \rangle$$

No hay una arista directa entre ningún v_i y v (para q' no haya un camino de longitud i)

$v_i \in V$; $\underline{k = |A_{i-1}|}$; $v_i \leq v_{i+1}$; $v \notin \{v_1, \dots, v_k\}$ y $\overbrace{v \notin \bigcup_{1 \leq i \leq k} E(v_i)}$. $(E(x) = \{y \in V : (x, y) \in E\})$

Certificado de que $|A_i| = a$ conociendo $|A_{i-1}|$

$$Z^{|A_{i-1}|}_{|A_i| = a} = \langle (1, Z_1), (2, Z_2), \dots, (n, Z_n) \rangle$$

Son los v, tal q' haya camino de s a v.

① Si $v \in A_i \Rightarrow Z_v = \underbrace{Z_{v \in A_i}}_{\text{logspace}}$; ② Si $v \notin A_i \Rightarrow Z_v = \underbrace{Z^{|A_i|-1}_{v \notin A_i}}_{\text{logspace}}$; $|\overbrace{v : Z_v = Z_{v \in A_i}}| = |A_i| = a$.

Verifica q' la cantidad de estos sea a.

La decisión de si poner ① ó ② en Z_i es responsabilidad del certificado. *'

Certificado final de que $t \notin A_n$

Es una lista de certificados tal que:

$$Z = \langle Z^{|A_0|}_{|A_1| = a_1}, Z^{|A_1|}_{|A_2| = a_2}, \dots, Z^{|A_{n-1}|}_{|A_n| = a_n}, Z^{|A_n|}_{t \notin |A_n|} \rangle$$

$a_i = |A_i|$; $A_0 = \{s\}$; $|A_0| = 1$.

El tamaño de Z es poly y q' el verificador puede chequear el certificado en espacio $O(\log(n))$

A medida q' lee el certificado de izq. a der., va guardando $|A_{i-1}|$ para verificar $Z^{|A_{i-1}|}_{|A_i| = a_i}$.

γ

Corolario 4. $\mathbf{NL} = \mathbf{coNL}$.

Demo:

Como sé q' un lenguaje coNL-completo ($\overline{PATH}$) pertenece a NL (por el anterior teorema), por lo cual puedo decir q' para cualquier $L \in$ coNL puedo reducirlo a un problema en NL. O sea, coNL $\subseteq$ NL.

Para ver q' NL $\subseteq$ coNL es muy similar.

Teorema 24. *Si* $S(n) \geq \log n$ *es construible en espacio entonces* $\mathbf{NSPACE}(S(n)) = \mathbf{coNSPACE}(S(n))$.

Observemos que $\mathbf{NL} = \mathbf{coNL}$ es un caso particular del Teorema 24 cuando $S(n) = \log n$.

→ P.ej: NSPACE(n^c) = coNSPACE(n^c), o sea NPSPACE = coNPSPACE. Esto lo sabía igual pq', PSPACE = NPSPACE.

Proposición 26. MAXINDSET $\in \Sigma_2^{\mathbf{P}}$.

Demo:

M máq. det. con entrada $x = \langle V, E, K, C, D\rangle$ ($G = (V,E)$), C y D son subconjuntos de V y K es un número.

$M : \langle x\rangle$

Retorno 1 si C, D son cjtos. independientes de G, C con K vértices y D con $\leq$ la cantidad de vértices de C.

Si no, retorno 0.

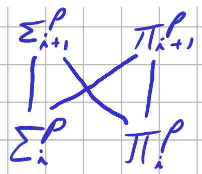
Luego, tenemos que:

$\underbrace{\langle \overbrace{V, E}^{G}, K\rangle}_{x} \in MAXINDSET$ si: $\exists C \in \{0,1\}^{|x|} . \forall D \in \{0,1\}^{|x|} . M(\langle V, E, K, C, D\rangle) = 1$

Proposición 27. *La jerarquía polinomial tiene las siguientes propiedades:*

$$\mathbf{\Sigma_1^P} = \mathbf{NP},\ \mathbf{\Sigma_i^P} \subseteq \mathbf{\Sigma_{i+1}^P},\ \mathbf{\Pi_i^P} \subseteq \mathbf{\Sigma_{i+1}^P},\ \mathbf{\Pi_1^P} = \mathbf{coNP},\ \mathbf{\Sigma_i^P} \subseteq \mathbf{\Pi_{i+1}^P},\ \mathbf{\Pi_i^P} \subseteq \mathbf{\Pi_{i+1}^P},\ \mathbf{PH} = \bigcup_{i \geq 0} \mathbf{\Sigma_i^P}$$

La pregunta $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ se puede generalizar a $\mathbf{\Sigma_i^P} \stackrel{?}{=} \mathbf{\Sigma_{i+1}^P}$; ninguna se conoce. Decimos que la jerarquía polinomial **colapsa** si $\mathbf{P} = \mathbf{PH}$. Veamos que si $\mathbf{P} = \mathbf{NP}$ entonces la jerarquía polinomial colapsa.



Proposición 28. *Si* $\mathbf{P} = \mathbf{NP}$ *entonces* $\mathbf{PH} = \mathbf{P}$.

Demo:

Por inducción:

Caso base: $\Sigma_1^P = \Pi_1^P$ → Sale por la suposición "Si P=NP".

Hip. Inductiva: $\Sigma_i^P, \Pi_i^P \subseteq P$

Caso inductivo:

Asumo q' u_j $(1 \leq j \leq i+1) \in \{0,1\}^{q(|x|)}$

QvQ $\Sigma^P_{i+1}, \Pi^P_{i+1} \subseteq P$.

Sea $\mathcal{L} \in \Sigma^P_{i+1}$. Existe M máq. det. poly y un polinomio q, tal que:

$$x \in \mathcal{L} \text{ sii } \underbrace{\exists u_1 \forall u_2 \cdots Q_{i+1} u_{i+1}.}_{i \text{ alternancias } (\Sigma^P_{i+1})} M(\langle x, u_1, u_2, \dots, u_{i+1} \rangle) = 1$$

Luego, definimos $\mathcal{L}'$ tal que:

$$\langle x, u_1 \rangle \in \mathcal{L}' \text{ sii } \underbrace{\forall u_2 \cdots Q_{i+1} u_{i+1}.}_{i-1 \text{ alternancias } (\Pi^P_i)} M(\langle x, u_1, u_2, \dots, u_{i+1} \rangle) = 1.$$

Después por HI sé q' $\mathcal{L}' \in \Pi^P_i \subseteq P$, por lo cual existe M' máq. det. poly. tal q':

$$\langle x, u_1 \rangle \in \mathcal{L}' \text{ sii } M'(\langle x, u_1 \rangle) = 1.$$

Lo cual implica que:

$$x \in \mathcal{L} \text{ sii } \overbrace{\exists u_1 . M'(\langle x, u_1 \rangle) = 1}^{\text{Def. de NP.}}$$

Por último como $\mathcal{L} \in NP = P$. Digo q' el $\mathcal{L}$ es arbitrario en Σ^P_{i+1}, llegamos a q' $\Sigma^P_{i+1} \subseteq P$.

Es análogo para $\Pi^P_{i+1} \subseteq P$.

γ

Proposición 29. *Las clases $\mathbf{\Sigma^P_i}$ y $\mathbf{\Pi^P_i}$ están cerradas hacia abajo por $\leq_p$. Es decir, para $\mathcal{L}' \leq_p \mathcal{L}$ tenemos que si $\mathcal{L} \in \mathbf{\Sigma^P_i}$ entonces $\mathcal{L}' \in \mathbf{\Sigma^P_i}$ y si $\mathcal{L} \in \mathbf{\Pi^P_i}$ entonces $\mathcal{L}' \in \mathbf{\Pi^P_i}$.*

Demo:

Cerradas para abajo: Si tengo $\mathcal{L}', \mathcal{L}$ tal q' $\mathcal{L}' \leq_p \mathcal{L}$ y $\mathcal{L} \in \Sigma^P_i \Rightarrow \mathcal{L}' \in \Sigma^P_i$.
Idem para Π^P_i.

Me sonaría raro q' no igual... ¿Hay algún lenguaje q' no?

Sea M máq. det. poly. y q un polinomio tal q' para todo $x \in \{0,1\}^*$

$$x \in \mathcal{L} \text{ sii } \exists u_1 \dots Q_i u_i . \underbrace{M(\langle f(x), u_1, \dots, u_i \rangle)}_{M'(\langle x, u_1, \dots, u_i \rangle)} = 1 \quad (u_j \in \{0,1\}^{q(|x|)} \text{ con } 1 \leq j \leq i).$$

Como f es computable en tiempo poly y M corre en poly, entonces M' corre en poly también.

γ

Proposición 30. *Si existe $\mathcal{L} \in$ **PH-completo** entonces existe i tal que $\mathbf{PH} = \mathbf{\Sigma^P_i}$.*

Demo:

$\mathcal{L} \in PH$ y $\mathcal{L} \in PH$-hard ($\mathcal{L}' \leq_p \mathcal{L}$ $\forall \mathcal{L}' \in PH$). Al ser q' $PH = \bigcup_{i \in \mathbb{N}} \Sigma^P_i$, existe i tal q' $\mathcal{L} \in \Sigma^P_i$.

Luego, qvq $PH \subseteq \Sigma_i^p$ (pq' $\supseteq$ es trivial). $\mathcal{L}' \in PH$, pero como $\mathcal{L}' \leq_p \mathcal{L} \Rightarrow \mathcal{L}' \in \Sigma_i^p$

Por prop. 29 sé q' si $\mathcal{L}' \leq_p \mathcal{L}$, entonces $\mathcal{L}' \in \Sigma_i^p$, por último $PH = \Sigma_i^p$.

Corolario 5. *Si la jerarquía polinomial no colapsa en ningún nivel entonces* $\mathbf{PH} \neq \mathbf{PSPACE}$.

Demo:

Por el contrarrecíproco,

Si $PH = PSPACE$ entonces existirían PH-completos (ya q' los hay PSPACE-completos)

Luego, por la Prop. 30 la jerarquía colapsaría.

Proposición 31. *Para todo* $i > 0$, $\Sigma_i\mathsf{SAT} \in \mathbf{\Sigma_i^P}$.

Demo:

Sea M máq. det. poly. q' con entrada $\langle \varphi, u_1, \dots, u_i \rangle$ hace lo siguiente:

Chequea q' φ es una QBF y q' $\forall_j . |u_j| \geq |\bar{y}_j|$.

Revisa q' $(u_1 \mapsto |y_1|) \dots (u_i \mapsto |y_i|) \models \varphi$.

$(K = |\langle \varphi \rangle|)$
$(u_j \in \{0,1\}^K)$

Luego, la M anterior corre en tiempo poly. y puedo ver q':

$$\langle \varphi \rangle \in \Sigma_i SAT \text{ sii } \underbrace{\exists u_1 . \forall u_2 \dots Q_i u_i}_{i-1 \text{ alternancias}} M(\langle \varphi, u_1, \dots, u_i \rangle)$$

Proposición 32. *Para todo* $i > 0$, $\Sigma_i\mathsf{SAT} \in \mathbf{\Sigma_i^P}$**-hard**.

Demo:

Símil al Teo. de Cook Levin. Ver.

Demostración. Recordemos la demostración del Teorema 11. De (10) y (11) sabemos que para todo $\mathcal{L} \in \mathbf{NP}$ existe una máquina determinística M que corre en tiempo polinomial t, un polinomio p y una fórmula booleana φ_x tal que para todo $x \in \{0,1\}^*$,

$$x \in \mathcal{L} \quad \text{sii} \quad \exists u \in \{0,1\}^{p(|x|)}\ M(xu) = 1 \quad \text{sii} \quad \varphi_x \in \mathsf{SAT}.$$

La fórmula booleana $\varphi_x = \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$ era computable en tiempo polinomial a partir de x, una vez que fijábamos M. Además, recordemos de (13) que cada ψ_j $(j = 1 \dots 4)$ tenía variables $p_0, \dots, p_{2n+3}$ $(n = |x| + p(|x|))$ para codificar la entrada de M (parte corresponde a x y parte al certificado u; se usaban dos variables para codificar cada uno de los símbolos $\{0, 1, \triangleright, \square\}$ que pueden aparecer en la cinta de entrada), y variables $q_1^j, \dots q_k^j$ (donde k dependía solo de M) para $j = 0, \dots, m = t(n)$ para codificar cada una de las mini-configuraciones $z_0, \dots, z_m$ del cómputo de M con entrada xu. Llamemos $\bar{e}$ a la tupla de variables booleanas $p_0, \dots, p_{2|x|+1}, p_{2n+2}, p_{2n+3}$, donde $p_0, \dots, p_{2|x|+1}$ codifica la porción inicial de la cinta de entrada conteniendo $\triangleright x$, y p_{2n+2}, p_{2n+3} codifica el símbolo $\square$ después de $\triangleright xu$. Llamemos $\bar{c}$ a la tupla de variables booleanas $p_{2|x|+2}, \dots, p_{2n+1}$, que codifica porción de la cinta de entrada que corresponde al certificado u. Por último, para cada $j = 1, \dots, m$, llamemos $\bar{q}_j$ a la tupla de variables booleanas $q_1^j, \dots q_k^j$, que codifica la mini-configuración z_j. Todas estas tuplas de variables booleanas tienen una dimensión polinomial en $|x|$. En concreto, ψ_1 expresaba que la entrada empezaba con x (fija el valor de las variables de $\bar{e}$ y menciona a las de $\bar{c}$ solo para codificar que $u \in \{0,1\}^*$), ψ_2 expresaba que z_0 es la configuración inicial, ψ_3 expresaba que z_j evolucionaba en un paso en z_{j+1} para $j = 0, \dots, m-1$, y ψ_4 expresaba que z_m era una configuración final aceptadora de M con entrada x. Lo importante es que

$$\begin{aligned} M(xu) = 1 \quad & \text{sii} \quad \tilde{u} \models \exists \bar{e}, \bar{q}_0, \dots, \bar{q}_m\ \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4 \\ & \text{sii} \quad \tilde{u} \models \forall \bar{e}, \bar{q}_0, \dots, \bar{q}_m\ (\psi_1 \wedge \psi_2 \wedge \psi_3) \rightarrow \psi_4, \end{aligned}$$

donde $|u| = q(|x|)$ y $\tilde{u} = u(0)u(0)u(1)u(1)\dots u(|u|-1)u(|u|-1)$ es la codificación de u y corresponde a la valuación para las variables de $\bar{c}$, las únicas variables libres de las fórmulas booleanas de arriba, que corresponden al certificado de M.

Esto fue solo un repaso de la demostración del Teorema 11. Veamos cómo usarlo ahora para probar la Proposición que estamos queriendo probar. Supongamos que $\mathcal{L} \in \mathbf{\Sigma_i^P}$. Sea M la máquina determinística oblivious, sin cinta de salida y con única cinta de trabajo (recordar la Proposición 3 y la Proposición 5), que corre en tiempo polinomial $t(n)$ y tal que para todo $x \in \{0,1\}^*$ tenemos que

$$x \in \mathcal{L} \quad \text{sii} \quad \underbrace{\exists u_1 \in \{0,1\}^{q(|x|)} \dots Q_i u_i \in \{0,1\}^{q(|x|)}}_{i-1 \text{ alternancias}}\ M(xu_1 \dots u_i) = 1.$$

Si $Q_i = \exists$ tenemos que $x \in \mathcal{L}$ sii $\models \rho_x$ sii $\rho_x \in \Sigma_i\mathsf{SAT}$, donde

$$\rho_x = \underbrace{\exists \bar{c}_1 \forall \bar{c}_2 \dots \overbrace{\exists \bar{c}_i \exists \bar{e}, \bar{q}_0, \dots, \bar{q}_m}^{\exists}}_{i-1 \text{ alternancias}}\ \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4.$$

Observemos que ρ_x es una QBF con todas las tuplas cuantificadas de tamaño polinomial en $|x|$ que se computa en tiempo polinomial a partir de x. Cada $\bar{c}_j$ es una tupla de variables booleanas de dimensión $2 \cdot q(|x|)$ y por lo tanto la tupla $\bar{c}$ descripta más arriba es $\bar{c}_1 \dots \bar{c}_i$, de dimensión $2 \cdot i \cdot q(|x|)$. Observemos también que el último bloque se convierte en un $\exists$, entonces ρ_x sigue teniendo $i-1$ alternancias de cuantificadores. Si $Q_i = \forall$ tenemos que $x \in \mathcal{L}$ sii $\models \rho_x$ sii $\rho_x \in \Sigma_i\mathsf{SAT}$, donde

$$\rho_x = \underbrace{\exists \bar{c}_1 \forall \bar{c}_2 \dots \overbrace{\forall \bar{c}_i \forall \bar{e}, \bar{q}_0, \dots, \bar{q}_m}^{\forall}}_{i-1 \text{ alternancias}}\ (\psi_1 \wedge \psi_2 \wedge \psi_3) \rightarrow \psi_4.$$

En este caso, el último bloque se convierte en un $\forall$, entonces ρ_x sigue teniendo $i-1$ alternancias de cuantificadores. Como la función $x \mapsto \rho_x$ es computable en tiempo polinomial, y $x \in \mathcal{L}$ sii $\rho_x \in \Sigma_i\mathsf{SAT}$ concluimos que $\mathcal{L} \leq_p \Sigma_i\mathsf{SAT}$. Como $\mathcal{L}$ lo tomamos arbitrario en $\mathbf{\Sigma_i^P}$ concluimos que $\Sigma_i\mathsf{SAT}$ es $\mathbf{\Sigma_i^P}$**-hard**. $\square$

Corolario 6. *Para todo $i > 0$, $\Sigma_i\mathsf{SAT} \in \mathbf{\Sigma_i^P}$-completo.*

Proposición 33. *Para todo $i > 0$, $\Pi_i\mathsf{SAT} \in \mathbf{\Pi_i^P}$-completo.*

Proposición 34. *Si $\mathcal{X} \in \mathbf{P}$, entonces $\mathbf{P} = \mathbf{P}^{\mathcal{X}}$.*

Demo:

Que $P \subseteq P^{\mathcal{X}}$ es trivial.

QvQ $P^{\mathcal{X}} \subseteq P$:

Sea M máq. det. poly. y $\mathcal{L} \in P^{\mathcal{X}}$ tal q' $\mathcal{X}(M)$. Luego, sea M' máq. det. poly con acceso al oráculo $\mathcal{X}$ q' decide $\mathcal{X}$.

Definimos M'' tal q' funciona como M' pero por cada llamada al oráculo con la consulta x llama a $M(x)$.

Como $M(x)$ es poly. respecto a x, lo es respecto a la entrada pq' si $|x| = n^c$, la entrada es de tamaño n y $|M(x)| = |x|^d$ entonces queda q' $|M(x)| = n^{c \cdot d}$, lo cual sigue siendo poly respecto a n.

Además como M'' funciona igual q' M', puedo decir q' $\mathcal{L}(M'')$ con M'' det. poly., o sea, $\mathcal{L} \in P$.

Proposición 35. $\textsc{ExpTime} \subseteq \mathbf{P}^{\mathsf{EXPCOM}}$.

Demo:

$\text{EXPCOM} \overset{\text{"}\pi\text{"}}{=} \{\langle M, x, 1^n\rangle :$ La máq. det. M con entrada x devuelve 1 en a lo sumo 2^n pasos$\}$

Sea $\mathcal{L} \in ExpTime = DTime(2^{n^c})$ para alguna ctte. c y supongamos q' $\mathcal{L}(M)$ con M una máq. det. q' corre en tiempo $O(2^{n^c}) = d.2^{n^c}$ con d ctte.

Luego, existe n_0 tal q' para todo $x \in \{0,1\}^*$, $|x| > k$ tenemos q' M con entrada x termina en a lo sumo $2^{|x|^{c+1}}$ pasos (esto viene de la def. de O)

Ahora consideramos M' máq. det. con entrada x y acceso al oráculo π, que hace lo siguiente:

Para los casos finitos $|x| \leq n_0$: Calculo manualmente y retorno si $x \in \mathcal{L}$

Para el resto, pregunto al oráculo si $\langle M, x, 1^{|x|^{c+1}}\rangle$ y retorno el resultado de esto.

M' corre en tiempo poly. y usa EXPCOM de oráculo, o sea, $\mathcal{L}(M^{\pi})$, por ende $\mathcal{L} \in P^{\pi}$

Proposición 36. $\mathbf{NP}^{\underset{\downarrow\,\pi}{\mathsf{EXPCOM}}} \subseteq \textsc{ExpTime}$.

Demo:

Sea $\mathcal{L} \in NP^{\pi}$ y $\mathcal{L}(N^{\pi})$ con N máq. no-det. q' corre en tiempo poly.

En tiempo exp. en $|x|$ puedo simular determinísticamente a N con entrada x (esto se ve como que $NP \subseteq E$) y también a cada consulta q' hace N al oráculo π (esto sale de q' las consultas son puestas en la cinta en tiempo poly., por lo cual, la ejecución de la máq. q' decide π tarda a lo suma $2^{O(|x|^c)}$ lo cual está en exp.).

Por ende, $NP^{\pi} \subseteq ExpTime$

Corolario 7. $\mathbf{P}^{\mathsf{EXPCOM}} = \mathbf{NP}^{\mathsf{EXPCOM}}$.

Demo:

Sale trivial que $P^{\pi} \subseteq NP^{\pi}$: Por la cadena de las 2 primeras $\subseteq$.

$$ExpTime \subseteq P^{\pi} \subseteq NP^{\pi} \subseteq ExpTime \subseteq NP^{\pi} \subseteq P^{\pi}$$

Por sanguche se puede ver q' $P^{\pi} = NP^{\pi}$

Teorema 25 (Baker, Gill, Solovay). *Existen oráculos $\mathcal{A}$ y $\mathcal{B}$ tal que* $\mathbf{P}^{\mathcal{A}} = \mathbf{NP}^{\mathcal{A}}$ *y* $\mathbf{P}^{\mathcal{B}} \neq \mathbf{NP}^{\mathcal{B}}$.

Antes de hablar de la demo, lo q' dice este teorema es que, $P \overset{?}{=} NP$ no se puede relativizar, o sea, no se puede usar la resolución de este enigma para análogamente resolver $P^{x} = NP^{x}$ para cualquier $x \subseteq \{0,1\}^*$.

Demo:

Sé que $P^{EXPCOM} = NP^{EXPCOM}$, entonces declaro $A = EXPCOM$.

Ahora quiero encontrar un B tal q' $P^{B} \neq NP^{B}$:

Def. de U_B, dado B : Para cualquier $B \subseteq \{0,1\}^*$ definimos: Galerazo ^^

$$U_B = \{1^n : \exists x \in B, |x| = n\}$$

U_B = Cjto. de palabras en unario con la longitud de alguna palabra en B.

$U_B \in NP^{B}$ para cualquier B lo demuestro haciendo N máq. no-det. con oráculo B y entrada y tal que:

Si y no sigue la forma 1^n : ret 0

Si no,

Inventa x tal q' $|x| = 1^n$

Consulta si $x \in B$: ret rta. de consulta.

Luego, tenemos q':

$$1^n \in \mathcal{L}(N^B) \text{ sii } \exists x \in B, |x| = n \text{ sii } 1^n \in U_B$$

O sea, $U_B \in NP^{B}$ por esto puntualmente y q' equivale a esto

Ahora, la parte difícil, definir un B tal q' $U_B \notin P^B$ (para ver la $\in NP^B$ no fue necesario, pero ahora si, pues si nunca lo hiciera, $P^B \neq NP^B$ para cualquier B).

¿Qué buscamos en B? ¿Por qué? Ni idea

Definiremos $B = \bigcup_i B_i$, y al mismo tiempo una sucesión $(n_i)_{i \in \mathbb{N}}$. Las prop. de B_i y n_i que buscamos son:

- $B_0 = \emptyset$ y $n_0 = 1$
- $B_i \subseteq B_{i+1}$ y $n_i < n_{i+1}$ — n_i aparenta ser un delimitador del tamaño de las palabras en B_i.
- $x \in B_i \Rightarrow |x| \leq n_i$ (en particular, cada B_i es finito)