

# Teoremas, prop, etc. (Mi punto de vista de demo)

Obs:  $L \subseteq NL \subseteq P$

Demo: Si  $L$  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 ← s; m ← 0
mientras m < |V|:
    z ← inventar un valor en {0, ..., |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  $q_{si}$ 
        si no:
            y ← z; m ← m + 1
            pasar a  $q_{no}$ 
```

→ Es útil saberse el algoritmo concreto, pg' es usado algo similar en muchas casas.

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

Demo:

Puedo reducir a cualquier  $L \in NL$  a un  $L \neq$  de los triviales.

Es absurdo pg' puedo computar  $X_L$  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  $Mg$  y cada vez q' que requiera un bit de  $f(x)$ , lo computo con  $M_f$ .

$Mg(\langle x, i \rangle) = g(x)(i)$   
 $M_f(\langle x, i \rangle) = f(x)(i)$

} Bit  $i$  de la salida de hacer  $g/f$  con entrada  $x$ .

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

Demo:

PATH E 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,

$x \in L$  si:  $N$  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 guarda 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  $(2^{c \cdot \log |x|})^2$

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

Usa 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  $L \in \text{NL}$ ,  $L \leq_L \text{PATH}$ , puedo decir q' puedo ver que para cualquier  $L \in \text{coNL}$  buscar  $L \in \text{NL}$  y reducirlo a  $\text{PATH}$ , donde su complemento equivaldría a  $L$ .

$x \in L \iff x \notin L \iff f(x) \notin \text{PATH} \iff f(x) \in \text{PATH}$

Esto es computable implícitamente en  $L$ , por lo cual:

$L \leq_L \text{PATH}$   $\uparrow\uparrow$

**Teorema 22.**  $L \in \text{NL}$  sii existe un polinomio  $p : \mathbb{N} \rightarrow \mathbb{N}$  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  $x \in \{0,1\}^*$ ,  $x \in L$  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-Szelepcsenyi:**

**Teorema 23 (Immerman-Szelepcsenyi).**  $\text{PATH} \in \text{NL}$ .

**Demo:**

$Q \vee Q \langle G, s, t \rangle \notin \text{PATH}$  se puede verificar en tiempo poly.

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

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

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

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

Luego q' la no pertenencia de  $t$  en  $A_n$  está en NL.

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

Partimos el certificado en distintos  $Z$  para distintos hechos.

Certificado de  $q' \forall v \in A_i$ :

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

$v_0 = s$  ;  $v_i (0 \leq i \leq K)$  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  $q' \forall v \notin A_i$  conociendo  $|A_i|$

Similar al anterior,

Todos son vdd.

En  $A_i$  se alcanzan exactamente  $|A_i|$  nodos

$$Z_{v \notin A_i}^{|\mathbb{A}_i|} = \langle (v_1, Z_{v \in A_i}), (v_2, Z_{v \in A_i}), \dots, (v_K, Z_{v \in A_i}) \rangle$$

$v_j \in V$  ;  $K = |A_i|$  ;  $v_i \leq v_{i+1}$  ;  $v \notin \{v_1, \dots, v_K\}$

$Z_{v \notin A_i}^{|\mathbb{A}_i|}$  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  $q' \forall v \notin A_i$  conociendo  $|A_{i-1}|$

Similar a la anterior,

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

↑ No hay una arista directa entre ningún  $v_i$  y  $v$  (pues q' no hay un camino de longitud 1)

$v_j \in V$  ;  $K = |A_{i-1}|$  ;  $v_i \leq v_{i+1}$  ;  $v \notin \{v_1, \dots, v_K\}$  y  $v \notin \bigcup_{i \leq K} E(v_i)$ . ( $E(x) = \{y \mid (x, y) \in E\}$ )

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

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

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

① Si  $v \in A_i \Rightarrow Z_v = Z_{v \in A_i}$  ; ② Si  $v \notin A_i \Rightarrow Z_v = Z_{v \notin A_i}$  ;  $|v : Z_v = Z_{v \in A_i}| = |A_i| = a$ .

Verificar 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:

$$Z = \langle Z_{|A_1|=a_1}^{|\mathbb{A}_1|}, Z_{|A_2|=a_2}^{|\mathbb{A}_2|}, \dots, Z_{|A_{n-1}|=a_{n-1}}^{|\mathbb{A}_{n-1}|}, Z_{t \notin A_n}^{|\mathbb{A}_n|} \rangle$$

$a_i = |A_i|$  ;  $A_0 = \{s\}$  ;  $|A_{n-1}| = L$ .

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  $Z_{|A_i|=a_i}^{|\mathbb{A}_i|}$ .

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

Demo:

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

Para ver  $\text{NL} \subseteq \text{coNL}$  es muy similar.

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

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

Lo Pej:  $\text{NSPACE}(n^c) = \text{coNSPACE}(n^c)$ , o sea  $\text{NPSPACE} = \text{coNPSPACE}$ . Esto lo sabía igual pq',  $\text{PSPACE} = \text{NPSPACE}$ .

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

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  $\neq$  un número).

$M : \langle x \rangle$

Retorna 1 si  $C, D$  son cjs. independientes de  $G$ ,  $C$  con  $K$  vértices y  $D$  con  $\{K\}$  cantidad de vértices de  $C$ .

Si no, retorna 0.

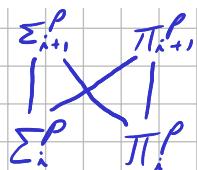
Luego, tenemos que:

$\langle \underbrace{\langle V, E, K \rangle}_{x} \rangle \in \text{MAXINDSET} \Leftrightarrow \exists C \in \{0, 1\}^{|V|}, \forall D \in \{0, 1\}^{|C|}, M(\langle V, E, K, C, D \rangle) = 1$

Proposición 27. La jerarquía polinomial tiene las siguientes propiedades:

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

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



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

Demo:

Por inducción:

Caso base:  $\Sigma_1^P = \Pi_1^P \rightarrow$  Sale por la suposición "Si:  $\text{P} = \text{NP}$ ".

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

Caso inductivo:

$$\forall i \in \mathbb{N} \quad \Sigma_{i+1}^P, \Pi_{i+1}^P \subseteq P.$$

Asumo q' u;  $(i; i+1) \in \{0, 1\}^{g'(x)}$

Sea  $L \in \Sigma_{i+1}^P$ . Existe  $M$  mág. det. poly y un polinomio  $g$ , tal que:

$x \in L \iff \exists u_1, u_2, \dots, u_{i+1} \in \{0, 1\}^{g(x)} \quad M(\langle x, u_1, u_2, \dots, u_{i+1} \rangle) = 1$

i alternancias ( $\Sigma_{i+1}^P$ )

Luego, definimos  $L'$  tal que:

$$\langle x, u \rangle \in L' \iff \exists u_1, u_2, \dots, u_{i+1} \in \{0, 1\}^{g(x)} \quad M(\langle x, u_1, u_2, \dots, u_{i+1} \rangle) = 1$$

i-1 alternancias ( $\Pi_i^P$ )

Después por HI sé q'  $L' \in \Pi_i^P \subseteq P$ , por lo cual existe  $M'$  mág. det. poly. tal q':

$\langle x, u \rangle \in L' \iff M'(\langle x, u \rangle) = 1$ .

Lo cual implica que: Def. de NP.

$x \in L \iff \exists u \in \{0, 1\}^{g(x)} \quad M'(\langle x, u \rangle) = 1$

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

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

**Proposición 29.** Las clases  $\Sigma_i^P$  y  $\Pi_i^P$  están cerradas hacia abajo por  $\leq_p$ . Es decir, para  $L' \leq_p L$  tenemos que si  $L \in \Sigma_i^P$  entonces  $L' \in \Sigma_i^P$  y si  $L \in \Pi_i^P$  entonces  $L' \in \Pi_i^P$ .

Demo:

Cerradas para abajo: Si tengo  $L', L$  tal q'  $L' \leq_p L$  y  $L \in \Sigma_i^P \Rightarrow L' \in \Sigma_i^P$ .  
Idem para  $\Pi_i^P$ .

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

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

$$x \in L \iff \exists u_1, \dots, u_i \in \{0, 1\}^{g(x)} \quad M(\langle f(x), u_1, \dots, u_i \rangle) = 1 \quad (u_i \in \{0, 1\}^{g(x)})$$

$$M'(\langle x, u_1, \dots, u_i \rangle)$$

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

Y

**Proposición 30.** Si existe  $L \in \text{PH-completo}$  entonces existe  $i$  tal que  $\text{PH} = \Sigma_i^P$ .

Demo:

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

Luego,  $\exists q \in PH \subseteq \Sigma_i^P$  ( $q \in P$  es trivial).

Por prop. 29 sé q' si:  $L' \leq_p L$ , entonces  $L' \in \Sigma_i^P$ , por último  $PH = \Sigma_i^P$ .

$L' \in PH$ , pero como  $L' \leq_p L \Rightarrow L' \in \Sigma_i^P$

✓

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

Demo:

Por el contrarrecíproco,

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

Luego, por la Prop. 30 la jerarquía colapsaría.

✓

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

Demo:

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

Chequea q'  $\varphi$  es una QBF y q'  $\models \varphi$ .

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

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

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

$\langle \varphi \rangle \in \Sigma_i \text{SAT} \text{ si: } \underbrace{\exists u_1 \forall u_2 \dots \exists u_i \forall u_{i+1} M(\langle \varphi, u_1, \dots, u_i \rangle)}$

$i-1$  alternancias

✓

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

Demo:

Simil al Teo. de Cook Levin. Ver.

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 valúación para las variables de  $\tilde{c}$ , 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  $L \in \Sigma_i^P$ . 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

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

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

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

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

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

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

$$M(xu) = 1 \quad \text{sii} \quad \tilde{u} \models \exists \tilde{c}, \tilde{q}_0, \dots, \tilde{q}_m \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

$$\text{sii} \quad \tilde{u} \models \forall \tilde{c}, \tilde{q}_0, \dots, \tilde{q}_m (\psi_1 \wedge \psi_2 \wedge \psi_3) \rightarrow \psi_4,$$

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

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

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

Demo:

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

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

Sea  $M$  máq. det. poly. y  $L \in \mathbf{P}^{\mathcal{X}}$  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  $pg'$  si  $|x| = n^c$ , la entrada es de tamaño  $n$  y  $|M(x)| = |x|^d$  entonces queda  $pg' |M(x)| = n^{c+d}$ , lo cual sigue siendo poly respecto a  $n$ .

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

Proposición 35.  $\text{EXPTime} \subseteq \mathbf{P}^{\text{EXPCom}}$ .

Demo:

$\text{EXPCom}^{\pi} = \{ \langle M, x, 1^n \rangle : M \text{ máq. det. con entrada } x \text{ devuelve } 1 \text{ en } \pi \text{ pasos} \}$

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

Luego, existe no tal q' para todo  $x \in \{0,1\}^*$ ,  $|x| > K$  tenemos q'  $M$  con entrada  $x$  termina en la sumo  $2^{\lfloor \log_2 d \rfloor + 1}$  pasos (esto viene de la def. de  $O$ )

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

Para los casos finitos  $|x| \leq n_0$ : Calcula manualmente y retorna si  $x \in L$

Para el resto, pregunta al oráculo si  $\langle M, x, 1^{\lfloor \log_2 d \rfloor + 1} \rangle$  y retorna el resultado de esto.

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

Proposición 36.  $\mathbf{NP}^{\text{EXPCom}} \subseteq \text{EXPTime}$ .

$\downarrow$   
 $\pi$

Demo:

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

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

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

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

Demo:

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

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

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

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

Antes de hablar de la demo, lo q' dice este teorema es que,  $P \stackrel{?}{=} NP$  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  $P^{\text{EXPCOM}} = NP^{\text{EXPCOM}}$ , entonces declaro  $A = \text{EXPCOM}$ .

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:

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

$U_B = \text{Cjto. de palabras en unario con la longitud de alguna palabra en } B$ .

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

Si  $y$  no sigue la forma  $L^n$ : ret 0

Sino,

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

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

Luego, tenemos q':

$L^n \in L(N^B)$  si:  $\exists x \in B, |x|=n$  si:  $L^n \in U_B$

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

Ahora, la parte difícil, definir un  $B$  tal q'  $U_0 \notin P^B$  (para ver q'  $\in NP^B$  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  $B = \bigcup B_i$ , y al mismo tiempo una sucesión  $(n_i)_{i \in \mathbb{N}}$ . Los prop. de  $B$  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 apunta 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)

La idea es diagonalizar y lograr q' para cada  $i$ ,  $M_i$  con oráculo  $B$  corre en tiempo polinomial, entonces toma la decisión equivocada cuando la entrada es  $1^{n_i}$ .

Si  $M_i(1^{n_i}) = 1 \Rightarrow$  En  $B$  no debe haber ninguna cadena de longitud  $n_i$  (o sea,  $1^{n_i} \notin U_0$ ).

Si:  $M_i(1^{n_i}) = 0 \Rightarrow$  En  $B$  debe haber alguna cadena de longitud  $n_i$  (por lo q',  $1^{n_i} \in U_0$ ).

Luego, veremos q' ninguna máq. q' corra en tiempo poly. decide  $U_0$ , de modo q'  $U_0 \notin P^B$ .  
Más específicamente:

- $M_i^{B_i}(1^{n_i}) = M_i^B(1^{n_i})$ : Si no paró la máq. con oráculo  $B_i$ , la misma no va a parar en el oráculo  $B$  (lo mismo con el estado de aceptación y de rechazo).
- $M_i^{B_i}(1^{n_i}) = 1$  en tiempo  $2^{n_i-1}$  sii  $B_i$  no contiene cadenas de tamaño  $n_i$  sii  $1^{n_i} \notin U_{B_i}$  (sii  $1^{n_i} \notin U_0$ ).

Construcción de  $B_i$  y  $n_i$  para  $i > 0$ .

$\forall K. \forall i \leq i$  tengo a  $n_K$  definido, junto a  $B_{i-1}$ . Sea  $l$  el máx de las longitudes consultado por  $M_K^{B_{i-1}}(1^{n_K})$  al tiempo  $2^{n_K-1}$  (o sea el máximo  $n_K$ )

Definimos  $n_i = l + \max(l, n_{i-1})$ . Para definir  $B_i$ , simulamos  $M_i$  con entrada  $1^{n_i}$  por  $2^{n_i-1}$  pasos.

- Si  $M_i$  consulta por un  $x$  con  $|x| < n_i$ , le respondemos lo mismo q' "x  $\in B_{i-1}$ ". Es decir, la hacemos pasar a q resp: si "x  $\in B_{i-1}$ " y a q resp: si "x  $\notin B_{i-1}$ ".
- Si consulta por  $|x| > n_i$ , le respondemos "no", es decir, la hacemos pasar a q resp: no.

Si  $M_i(1^{n_i}) = 1$  en a lo sumo  $2^{n_i-1}$  pasos, definimos  $B_i = B_{i-1}$ .

En este caso  $B_i$  no contiene cadenas de longitud  $n_i$ , pq' la única oportunidad q' tenía de entrar a  $B$  era en el paso  $i$  de la construcción.

Si  $M_i(1^{n_i}) = 0$  en a lo sumo  $2^{n_i-1}$  pasos o no llegó a la decisión todavía, elegimos un  $x \in \{0, 1\}^*$  tal q'  $|x| = n_i$ , q' no haya sido consultado y definimos  $B_i = B_{i-1} \cup \{x\}$  (tal x existe pq' hay  $2^{n_i-1}$  números representables con  $n_i$  bits y solo hacemos  $2^{n_i-1}$  consultas).

Entonces,  $M_K(1^{n_K}) = 1$  sii  $B$  no contiene cadenas de longitud  $n_K$ .

Con este  $B$  definido, queremos:

Verificación de  $q'$ :  $U_B \notin P^B$

Sea  $M$  det. y  $p$  un polinomio tal que  $M^B$  corre en tiempo  $p(n)$  y  $M^B$  decide  $U_B$ .  
Sea  $i$  suficientemente grande tal que  $M_i = M$  y  $2^{n_i} > p(n_i)$ . Si ahora  $M_i = M$  por  $2^{n_i}$  pasos es suficiente para saber si  $M_i$  acepta o rechaza  $1^{n_i}$ .

Si  $M_i^B(1^{n_i}) = 1 \Rightarrow 1^{n_i} \notin U_B$   
Si  $M_i^B(1^{n_i}) = 0 \Rightarrow 1^{n_i} \in U_B$

} Esto es lo clave.

Por último,  $M^B = M_i^B$  no puede decidir  $U_B$  pq' falla para la entrada  $1^{n_i}$ .

Relación entre jerarquía polinomial y las clases NP

Teorema 26. Para  $i \geq 1$ ,  $\Sigma_{i+1}^P = NP^{\Sigma_i SAT}$ .

Y

Demo:

$\Sigma_{i+1}^P \subseteq NP^{\Sigma_i SAT}$ :

Sea  $L \in \Sigma_{i+1}^P$ , existe  $M$  det. poly. y un polinomio  $q$  tal que  $q'$ :

$x \in L \iff \exists \underbrace{u_1, u_2, \dots, u_{i+1} \in Q_{i+1} \cup \{1\}}_{i-1 \text{ alternancias}}. M(\langle x, u_1, u_2, \dots, u_{i+1} \rangle) = 1$

$(u_1, \dots, u_{i+1} \in \{0, 1\}^{q(x)})$

Definimos:

$L' = \{ \langle x, u_1 \rangle : \underbrace{\forall u_2, \dots, u_{i+1} \in Q_{i+1} \cup \{1\}}_{i-1 \text{ alternancias}}. M(\langle x, u_1, u_2, \dots, u_{i+1} \rangle) = 1 \}$

Como  $L' \in \Pi_{i+1}^P$ , sé que  $L' \not\subseteq \Sigma_i SAT$ .

Luego, definimos una máq. no-det.  $N$  tal que  $q'$  con el oráculo  $L'$  y entrada  $x \in \{0, 1\}^*$  hace:

Invento  $u_1$

Consulto si  $\langle x, u_1 \rangle \in L'$  y devuelvo la respuesta.

$L(N^L) = L$  se ve pq' resuelve  $q'$  existe  $u_1$  que cumple lo de la QBF  $q'$  arranca con  $\forall$ .

El oráculo  $L'$  cumple  $q'$   $L' \not\subseteq \Sigma_i SAT$ . Entonces,  $L \in NP^{\Sigma_i SAT} = NP^{\Sigma_i SAT}$

↳  $q'$  es invertir la salida de la rta. y toma  $O(1)$ .

$NP^{\Sigma_i SAT} \subseteq \Sigma_{i+1}^P$ : (Esto es un bando T.T)

Sea  $L \in NP^{\Sigma_i SAT}$  y  $N$  una máq. no-det. poly. en tiempo  $t$  tal que  $q'$  con oráculo  $\Sigma_i SAT$  decide  $L$ .

$x \in L \iff \exists \text{ cálculo } u \text{ de } N^{\Sigma_i SAT} \text{ con entrada } x \text{ q' llega a q'}$ .

A lo largo de  $u$ ,  $N$  hace consultas  $\varphi_1, \dots, \varphi_K$  (con  $K \leq t(|x|)$ ) al oráculo, del tipo

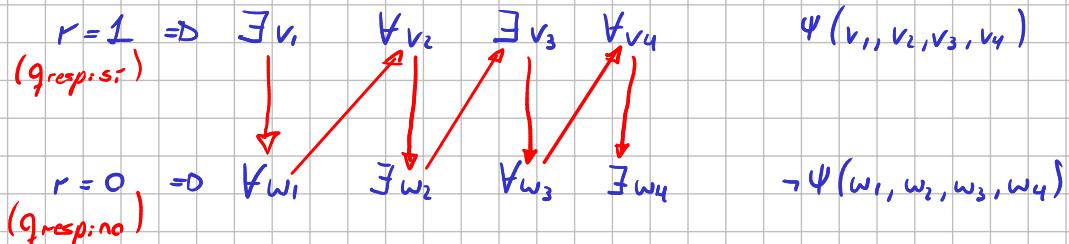
$\varphi_i = \exists \underbrace{u_1, u_2, \dots, u_i}_{i-1 \text{ alternancias}}. \psi(u_1, \dots, u_i)$

↳ Es de este estilo pq' es a  $\Sigma_i SAT$

donde  $\Psi_j (j=1, \dots, K)$  es una fórmula booleana y los  $\bar{w}_i$  son tuplas de variables booleanas, y recibe respuesta  $r_j \in \{0, 1\}$  ("sí" =  $q_{\text{resp},j} = 1$  ó "no" =  $q_{\text{resp},j} = 0$ )

- Si  $r_j = 1$  : Existe  $\bar{v}_i$  tal q'  $\bar{v}_i \models \bar{v}_i \dots Q \bar{v}_i \cdot \Psi_j(\bar{v}_1, \bar{v}_2, \dots, \bar{v}_i)$
- Si  $r_j = 0$  : Para todo  $\bar{w}_i$  tenemos  $\bar{w}_i \models \bar{v}_i \dots Q \bar{w}_i \cdot \Psi_j(\bar{w}_1, \bar{w}_2, \dots, \bar{w}_i)$ , o sea  $\bar{w}_i \models \exists \bar{v}_i \dots Q \bar{v}_i \rightarrow \Psi_j(\bar{w}_1, \bar{w}_2, \dots, \bar{w}_i)$

Ejemplo visual:



Entonces,

$$x \in L \text{ si: } \underbrace{\exists u \exists r \exists v \forall w, \forall v_1 \exists w_1 \exists v_2 \forall w_2 \exists v_3 \forall w_3 \forall v_4 \exists w_4}_{4 \text{ alternancias, o sea, } \Sigma_5^r}$$

Fin ejemplo visual

$x \in L$  si: Existe un cálculo  $u$ , variables booleanas  $(r_j)_{j=1, \dots, K}$  y tuplas booleanas  $(\bar{v}_i)_{i=1, \dots, K}$  y  $(\bar{w}_i)_{i=1, \dots, K}$  tal q'  $N$  acepta  $x$  siguiendo el cálculo  $u$ , recibe resp.  $r_1, \dots, r_K$  (en orden) a las consultas del oráculo para  $j=1, \dots, K$ .

- $r_j = 1$  y  $\bar{v}_i \models \forall \bar{v}_i \exists \bar{v}_3 \dots Q \bar{v}_i \cdot \Psi_j(\bar{v}_1, \bar{v}_2, \dots, \bar{v}_i)$  para algún  $\bar{v}_i$ .
- $r_j = 0$  y  $\bar{w}_i \models \exists \bar{w}_i \forall \bar{w}_3 \dots Q \bar{w}_i \cdot \neg \Psi_j(\bar{w}_1, \bar{w}_2, \dots, \bar{w}_i)$  para todo  $\bar{w}_i$ .

Luego,

$$x \in L \text{ si: } \underbrace{\exists u, (r_j), (\bar{v}_i), \forall (\bar{w}_i), \forall (\bar{v}_i), \exists (\bar{w}_i), \exists (\bar{v}_3), \forall (\bar{w}_3), \dots, Q(\bar{v}_i), Q(\bar{w}_i)}_{i-2 \text{ alternancias}}; \Psi$$

$\exists$        $\forall$        $i-2 \text{ alternancias}$

$i$  alternancias.

donde  $\Psi$  expresa q'  $N$  acepta  $x$  siguiendo el cálculo  $u$ , recibe como resp.  $r_1, \dots, r_K$  (en orden) a las consultas al oráculo y para  $j=1, \dots, K$ :

$$r_j = 1 \text{ y } \models \Psi_j(\bar{v}_1, \bar{v}_2, \dots, \bar{v}_i) \quad \text{o bien}$$

$$r_j = 0 \text{ y } \models \neg \Psi_j(\bar{w}_1, \bar{w}_2, \dots, \bar{w}_i)$$

Se ve q'  $\Psi$  expresa una condición q' es computable en tiempo poly.

Concluimos q' :

$$L \in \Sigma_{i+1}^P$$

✓

Teorema 27.  $P \subseteq P_{/\text{poly}}$ .

Demo:

Sea  $L \in P$  y  $M$  una máq. det. q' corre en tiempo  $p(n)$  con  $p$  un polinomio.

$x \in L$  si:  $M(x) = 1$  para todo  $x \in \{0,1\}^*$

Sea  $M$  oblivious, sin cinta de salida y con única cinta de trabajo. Luego,

$x \in L$  si: Existe una secuencia de mini-configurationes  $z_0, \dots, z_{p(x)}$  de  $M$  con entrada  $x$  tal q'  $z_0$  es inicial para  $x$ ,  $z_{p(x)}$  es final

$z_i$  se representa con la cadena  $e \uparrow$  ong!  $z_i$ :

$e \in \{0,1\}^2$  codifica el símbolo leído por la cabeza en la cinta de entrada

$t \in \{0,1\}^2$  " de trabajo

$s \in \{0,1\}^c$  codifica el estado de  $M$  (c es ctte.)

$$|z_i| = 4 + c = K.$$

Para  $n$  fijo, construimos  $C_n$  tal que  $M(x) = C_n(x)$  para todo  $x \in \{0,1\}^n$ .

$z_{i+1}$  depende de:

- $e$
- $z_i$
- $z_{\text{prev}(i,n)}$  (Anterior vez q' la cabeza de la cinta de trabajo estaba en la posición  $i$ )

Sea  $F_n: \{0,1\}^{2K+2} \rightarrow \{0,1\}^K$  tal q':

$$F_n(e \uparrow z_i \uparrow z_{\text{prev}(i,n)}) = z_{i+1}$$

Podemos representar  $F_n$  con un circuito de tamaño constante, independiente de  $n$ .

- Al ser  $M$  oblivious: Las pos. de la cabeza de entrada y trabajo de  $M$  solo dependen de  $n$  y del nro de paso del círculo de  $M$  con entrada  $x$ .

$$|C_n| = O(n + p(n)) \text{ y } C_n(x) = M(x), \text{ o sea } P \subseteq P_{\text{poly}}.$$

↑

Teorema 28.  $P \not\subseteq P_{\text{poly}}$ .

Demo:

Si  $L \subseteq \{1^n : n \in \mathbb{N}\}$ , entonces  $L \in P_{\text{poly}}$ .

O sea, puedo tomar un lenguaje unario  $L$  indecidible, por ejemplo:

$$H = \{1^n : x \text{ es la } n\text{-ésima cadena y } \text{last}(x) = 1\}$$

↑

Ordenamos las cadenas en orden lexicográfico y por longitud (las binarias, así las puedo representar unariamente)

Entonces, tenemos q'  $H \in P_{\text{poly}} \setminus P$  (ya q' en  $P$  no hay  $L$ 's indecidibles).

Y

Notación:  $C_n(\varphi) = C_n(\chi_{\text{SAT}}(\varphi))$  (idem para  $C'_n$ ).

**Proposición 37.** Sea  $(C_n)_{n \in \mathbb{N}}$  una familia de circuitos de tamaño  $S(n)$  tal que<sup>14</sup>  $C_n(\varphi) = \chi_{\text{SAT}}(\varphi)$  para toda fórmula booleana  $\varphi = \varphi(x_1, \dots, x_n)$  con  $|\varphi| = n$ . Entonces existe una familia de circuitos  $(C'_n)_{n \in \mathbb{N}}$  de tamaño polinomial en  $S(n)$  tal que para todo  $n$ :  $C'_n$  tiene  $n$  salidas y para toda fórmula booleana  $\varphi = \varphi(x_1, \dots, x_n)$  con  $|\varphi| = n$ , tenemos  $\varphi(C'_n(\varphi)) = \chi_{\text{SAT}}(\varphi)$ .

Lo sea, se puede hacer una  $(C'_n)_{n \in \mathbb{N}}$  q' adivine la valación  $q'$  que satisface una fórmula, si es  $q'$  existe.

Demo:

Si:  $\varphi \in \text{SAT}$ , entonces:

Reemplazo  $x_i$  con 1, si sigue siendo SAT continúo, si no a  $x_i$  le asigna 0.

↳  $\varphi(\chi_{\text{SAT}}(\varphi(1, x_2, \dots, x_n)), x_2, \dots, x_n) \in \text{SAT}$ .

Se repite el razonamiento con las demás variables.

Ahora q'q esto se pueda hacer en un circuito.

Dada  $\varphi = \varphi(x_1, \dots, x_n)$  de tamaño  $n$  y  $C_n$  tal q'  $C_n(\varphi) = \chi_{\text{SAT}}(\varphi)$ , definimos  $C'_n$  con entrada  $\varphi$  y salidas  $r_1, \dots, r_n$  ( $r_i \in \{0, 1\}$ ) como:

- $r_1 = C_n(\varphi(1, x_2, \dots, x_n))$
- $r_2 = C_n(\varphi(r_1, 1, \dots, x_n))$
- ...
- $r_n = C_n(\varphi(r_1, r_2, \dots, 1))$ .

↳ la valación

Luego, tenemos q'  $\varphi \in \text{SAT}$  si:  $r_1 \dots r_n \models \varphi$  si:  $\models \varphi(C'_n(\varphi))$

La evalúa.

✓

## Teorema de Karp-Lipton

**Teorema 29** (Karp-Lipton). Si  $\text{NP} \subseteq \text{P/poly}$ , entonces  $\text{PH} = \Sigma_2^{\text{P}}$ .

Demo: