Complejidad Computacional

Santiago Figueira

Departamento de Computación - FCEN - UBA

clase 9

Clase 9

La jerarquía polinomial

Problemas $\mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$**-completos**

La jerarquía polinomial

Clase 9

- La jerarquía polinomial
- Problemas $\mathbf{\Sigma_i^p}$**-completos**

Problemas **NP**

Problema: SAT (satisfacción booleana en CNF)

$$\mathsf{SAT} = \{\langle\varphi\rangle \colon \varphi \in \text{CNF es satisfacible}\}$$

Problemas **NP**

Problema: SAT (satisfacción booleana en CNF)

$$\mathsf{SAT} = \{\langle\varphi\rangle : \varphi \in \text{CNF es satisfacible}\}$$

$$\langle\varphi\rangle \in \mathsf{SAT} \qquad \text{sii} \qquad \exists v \quad \underbrace{v \models \varphi}_{\text{"}v\text{ es una valuación que satisface }\varphi\text{"} \rightsquigarrow \text{ polinomial}}$$

Problemas NP

Problema: SAT (satisfacción booleana en CNF)

$$\mathsf{SAT} = \{\langle\varphi\rangle : \varphi \in \text{CNF es satisfacible}\}$$

$$\langle\varphi\rangle \in \mathsf{SAT} \qquad \text{sii} \qquad \exists v \quad \underbrace{v \models \varphi}_{\text{“}v \text{ es una valuación que satisface } \varphi\text{”} \rightsquigarrow \text{ polinomial}}$$

Problema: Conjunto independiente

$$\mathsf{INDSET} = \{\langle G, k\rangle \mid G \text{ tiene un conjunto independiente de} \geq k \text{ vértices}\}$$

Problemas **NP**

Problema: SAT (satisfacción booleana en CNF)

$$\mathsf{SAT} = \{\langle\varphi\rangle : \varphi \in \text{CNF es satisfacible}\}$$

$$\langle\varphi\rangle \in \mathsf{SAT} \quad \text{sii} \quad {\color{red}\exists v} \quad \underbrace{v \models \varphi}_{\text{“}v\text{ es una valuación que satisface }\varphi\text{”} \rightsquigarrow \text{ polinomial}}$$

Problema: Conjunto independiente

$$\mathsf{INDSET} = \{\langle G, k\rangle \mid \text{G tiene un conjunto independiente de $\geq k$ vértices}\}$$

$$\langle\underbrace{(V,E)}_{G}, k\rangle \in \mathsf{INDSET} \text{ sii } {\color{red}\exists C} \underbrace{C \subseteq V, |C| = k, \neg\exists u, v \in C, (u,v) \in E}_{\text{“C es un conjunto independiente de G de k vértices”} \rightsquigarrow \text{ polinomial}}$$

Problemas coNP

Recordar

Problema: Tautología

$$\mathsf{TAUT} = \{\langle\varphi\rangle : \varphi \in \text{CNF es una tautología}\}$$

$$\langle\varphi\rangle \in \mathsf{TAUT} \qquad \text{sii} \qquad \forall v \quad \underbrace{v \models \varphi}_{\text{“}v\text{ es una valuación que satisface }\varphi\text{”} \rightsquigarrow \text{ polinomial}}$$

Otros problemas

Problema: Conjunto independiente máximo

$$\mathsf{MAXINDSET} = \{\langle G, k\rangle : \text{el conjunto independiente más grande de } G \text{ tiene } k \text{ vértices}\}$$

¿Podemos esto? $\langle G, k\rangle \in \mathsf{MAXINDSET}$ sii $\exists \underbrace{\ldots\ldots}_{\text{polinomial}}$

¿Y esto? $\langle G, k\rangle \in \mathsf{MAXINDSET}$ sii $\forall \underbrace{\ldots\ldots}_{\text{polinomial}}$

Otros problemas

Problema: Conjunto independiente máximo

$$\mathsf{MAXINDSET} = \{\langle G, k\rangle : \text{el conjunto independiente más grande de } G \text{ tiene } k \text{ vértices}\}$$

¿Podemos esto? $\langle G, k\rangle \in \mathsf{MAXINDSET}$ sii $\exists \underbrace{\ldots\ldots}_{\text{polinomial}}$

¿Y esto? $\langle G, k\rangle \in \mathsf{MAXINDSET}$ sii $\forall \underbrace{\ldots\ldots}_{\text{polinomial}}$

Pero sí podemos hacer esto:

$$\langle \overbrace{(V, E)}^{G}, k\rangle \in \mathsf{MAXINDSET} \quad \text{sii}$$

$$\exists C\ \forall D\ \underbrace{\begin{pmatrix} C \subseteq V, |C| = k \wedge \neg\exists u, v \in C \wedge (u, v) \in E \wedge \\ (D \subseteq V \wedge \neg\exists u, v \in D \wedge (u, v) \in E) \rightarrow |D| \leq |C| \end{pmatrix}}_{\text{“}C\text{ es un conjunto independiente de }G\text{ de }k\text{ vértices y si }D\text{ es un conjunto independiente de }G\text{, no puede ser más grande que }C\text{”} \rightsquigarrow \text{ polinomial}}$$

La jerarquía polinomial

Clase de complejidad: $\boldsymbol{\Sigma_i^p}$, $\boldsymbol{\Pi_i^p}$

- Para $i > 0$, $\boldsymbol{\Sigma_i^p}$ es la clase de lenguajes $\mathcal{L}$ tales que existe una máquina determinística M que corre en tiempo polinomial y un polinomio q tal que

$$
\begin{aligned}
x \in \mathcal{L} \qquad \text{sii} \qquad & \exists u_1 \in \{0,1\}^{q(|x|)} \\
& \forall u_2 \in \{0,1\}^{q(|x|)} \\
& \qquad\quad \vdots \\
& Q_i u_i \in \{0,1\}^{q(|x|)} \ M(\langle x, u_1, \ldots, u_i \rangle) = 1
\end{aligned}
$$

  donde $Q_i = \begin{cases} \forall & \text{si } i \text{ es par} \\ \exists & \text{si } i \text{ es impar} \end{cases}$

- $\boldsymbol{\Sigma_0^p} = \mathbf{P}$
- $\mathbf{PH} = \bigcup_{i \geq 0} \boldsymbol{\Sigma_i^p}$ es la **jerarquía polinomial**
- Para $i \geq 0$, $\boldsymbol{\Pi_i^p} = \{\overline{\mathcal{L}} : \mathcal{L} \in \boldsymbol{\Sigma_i^p}\}$

Alternancia de cuantificadores

Lo que importa en $\boldsymbol{\Sigma_i^{\mathbf{p}}}$ [resp. $\boldsymbol{\Pi_i^{\mathbf{p}}}$] es que la fórmula empiece con $\exists$ [resp. $\forall$] y haya $i-1$ *alternancias* de cuantificadores.
Bloques $\exists\exists$ se pueden unir en un solo $\exists$

Ejemplo

$$\begin{array}{l} \exists u_1 \in \{0,1\}^{q(|x|)} \\ \forall u_2 \in \{0,1\}^{q(|x|)} \\ \exists u_3 \in \{0,1\}^{q(|x|)} \\ \exists u_4 \in \{0,1\}^{q'(|x|)} \\ \quad M(\langle x, u_1, u_2, u_3, u_4\rangle) = 1 \end{array} \qquad \text{sii} \qquad \begin{array}{l} \exists u_1 \in \{0,1\}^{q(|x|)+q'(|x|)} \\ \forall u_2 \in \{0,1\}^{q(|x|)+q'(|x|)} \\ \exists u_3' \in \{0,1\}^{q(|x|)+q'(|x|)} \\ \quad M'(\langle x, u_1, u_2, u_3'\rangle) = 1 \end{array}$$

- M' solo lee los primeros $q(|x|)$ bits de u_1, u_2 (el resto los ignora)
- la información de u_3 y u_4 ahora la tiene pegada en $u_3' = u_3u_4$

Lo mismo vale para bloques $\forall\forall$.

Observaciones de la definición de $\boldsymbol{\Sigma_i^{\mathrm{p}}}$

$$
\begin{aligned}
x \in \mathcal{L} \qquad \text{sii} \qquad & \exists u_1 \in \{0,1\}^{q(|x|)} \\
& \forall u_2 \in \{0,1\}^{q(|x|)} \\
& \qquad \vdots \\
& Q_i u_i \in \{0,1\}^{q(|x|)} \; M(\langle x, u_1, \ldots, u_i \rangle) = 1
\end{aligned}
$$

- podemos pedir $M(xu_1 \ldots u_i) = 1$ en vez de $M(\langle x, u_1, \ldots, u_i \rangle) = 1$.
- podemos pedir distintos polinomios $q_1, \ldots, q_i$ para las longitudes de $u_1 \ldots, u_i$ en vez del mismo q para todos.

Ejemplo de problema en $\mathbf{\Sigma_2^p}$

Proposición

$\textsf{MAXINDSET} \in \mathbf{\Sigma_2^p}$.

Demostración.

Existe una máquina determinística M que corre en tiempo polinomial tal que

$$M(\langle \overbrace{V, E}^{G}, k, C, D\rangle) = 1 \quad \text{sii} \quad \begin{array}{l}C \text{ es un conjunto independiente} \\ \text{de } G \text{ de } k \text{ vértices y, si } D \text{ es un} \\ \text{conjunto independiente de } G\text{, el} \\ \text{tamaño de } C \text{ es mayor o igual al} \\ \text{tamaño de } D\end{array}$$

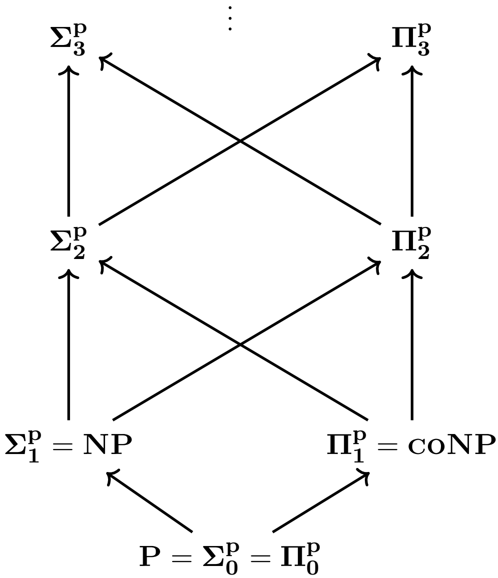
$$\underbrace{\langle \overbrace{(V, E)}^{G}, k\rangle}_{x} \in \textsf{MAXINDSET} \quad \text{sii} \quad \exists C\ \forall D\ M(\langle \underbrace{V, E, k}_{x}, C, D\rangle) = 1$$

Suponemos que $V = \{0, \ldots, n-1\}$. Los conjuntos C y D los codificamos con palabras en $\{0,1\}^{|V|}$. Como $|V| \leq |x|$, podemos codificar C, D como palabras de $\{0,1\}^{|x|}$. □

Propiedades de la jerarquía polinomial

Proposición

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



¿$\boldsymbol{\Sigma}_i^{\mathbf{p}} \subsetneq \boldsymbol{\Sigma}_{i+1}^{\mathbf{p}}$?

$\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ se puede generalizar a $\boldsymbol{\Sigma}_i^{\mathbf{p}} \stackrel{?}{=} \boldsymbol{\Sigma}_{i+1}^{\mathbf{p}}$; ninguna se conoce.

Proposición

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

Ejercicio

Para todo $i > 0$, si $\boldsymbol{\Sigma}_i^{\mathbf{p}} = \boldsymbol{\Pi}_i^{\mathbf{p}}$ entonces $\mathbf{PH} = \boldsymbol{\Sigma}_i^{\mathbf{p}}$.

Demostración de $\mathbf{P} = \mathbf{NP} \Rightarrow \mathbf{PH} = \mathbf{P}$.

Sup. $\mathbf{P} = \mathbf{NP}$. Probamos $\boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}, \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}} \subseteq \mathbf{P}$ por inducción en $i \geq 1$.

Demostración de $\mathbf{P} = \mathbf{NP} \Rightarrow \mathbf{PH} = \mathbf{P}$.

Sup. $\mathbf{P} = \mathbf{NP}$. Probamos $\boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}, \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}} \subseteq \mathbf{P}$ por inducción en $i \geq 1$.
$i = 1$: trivial porque $\mathbf{P} = \mathbf{NP} = \boldsymbol{\Sigma}_{\mathbf{1}}^{\mathbf{p}}$ y $\mathbf{P} = \mathbf{coNP} = \boldsymbol{\Pi}_{\mathbf{1}}^{\mathbf{p}}$.

Demostración de $\mathbf{P} = \mathbf{NP} \Rightarrow \mathbf{PH} = \mathbf{P}$.

Sup. $\mathbf{P} = \mathbf{NP}$. Probamos $\boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}, \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}} \subseteq \mathbf{P}$ por inducción en $i \geq 1$.
$i = 1$: trivial porque $\mathbf{P} = \mathbf{NP} = \boldsymbol{\Sigma}_{\mathbf{1}}^{\mathbf{p}}$ y $\mathbf{P} = \mathbf{coNP} = \boldsymbol{\Pi}_{\mathbf{1}}^{\mathbf{p}}$.
Sup. $\boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}, \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}} \subseteq \mathbf{P}$ y probemos $\boldsymbol{\Sigma}_{\boldsymbol{i+1}}^{\mathbf{p}}, \boldsymbol{\Pi}_{\boldsymbol{i+1}}^{\mathbf{p}} \subseteq \mathbf{P}$. Sea $\mathcal{L} \in \boldsymbol{\Sigma}_{\boldsymbol{i+1}}^{\mathbf{p}}$.

Demostración de $\mathbf{P} = \mathbf{NP} \Rightarrow \mathbf{PH} = \mathbf{P}$.

Sup. $\mathbf{P} = \mathbf{NP}$. Probamos $\mathbf{\Sigma_i^p}, \mathbf{\Pi_i^p} \subseteq \mathbf{P}$ por inducción en $i \geq 1$.
$i = 1$: trivial porque $\mathbf{P} = \mathbf{NP} = \mathbf{\Sigma_1^p}$ y $\mathbf{P} = \mathbf{coNP} = \mathbf{\Pi_1^p}$.
Sup. $\mathbf{\Sigma_i^p}, \mathbf{\Pi_i^p} \subseteq \mathbf{P}$ y probemos $\mathbf{\Sigma_{i+1}^p}, \mathbf{\Pi_{i+1}^p} \subseteq \mathbf{P}$. Sea $\mathcal{L} \in \mathbf{\Sigma_{i+1}^p}$.
Existe una máquina determinística M que corre en tiempo polinomial y un polinomio q tal que

$$x \in \mathcal{L} \quad \text{sii} \quad \begin{array}{l} \exists u_1 \in \{0,1\}^{q(|x|)}\ \forall u_2 \in \{0,1\}^{q(|x|)} \dots \\ \cdots Q_{i+1} u_{i+1} \in \{0,1\}^{q(|x|)}\ M(\langle x, u_1, \dots u_{i+1} \rangle) = 1 \end{array}$$

Demostración de $\mathbf{P} = \mathbf{NP} \Rightarrow \mathbf{PH} = \mathbf{P}$.

Sup. $\mathbf{P} = \mathbf{NP}$. Probamos $\mathbf{\Sigma_i^p}, \mathbf{\Pi_i^p} \subseteq \mathbf{P}$ por inducción en $i \geq 1$.
$i = 1$: trivial porque $\mathbf{P} = \mathbf{NP} = \mathbf{\Sigma_1^p}$ y $\mathbf{P} = \mathbf{coNP} = \mathbf{\Pi_1^p}$.
Sup. $\mathbf{\Sigma_i^p}, \mathbf{\Pi_i^p} \subseteq \mathbf{P}$ y probemos $\mathbf{\Sigma_{i+1}^p}, \mathbf{\Pi_{i+1}^p} \subseteq \mathbf{P}$. Sea $\mathcal{L} \in \mathbf{\Sigma_{i+1}^p}$.
Existe una máquina determinística M que corre en tiempo polinomial y un polinomio q tal que

$$x \in \mathcal{L} \quad \text{sii} \quad \begin{array}{l} \exists u_1 \in \{0,1\}^{q(|x|)}\ \forall u_2 \in \{0,1\}^{q(|x|)} \dots \\ \cdots Q_{i+1} u_{i+1} \in \{0,1\}^{q(|x|)}\ M(\langle x, u_1, \dots u_{i+1} \rangle) = 1 \end{array}$$

Definimos $\mathcal{L}'$ así: $\langle x, u_1 \rangle \in \mathcal{L}'$ sii

$$\forall u_2 \in \{0,1\}^{q(|x|)} \cdots Q_{i+1} u_{i+1} \in \{0,1\}^{q(|x|)}\ M(\langle x, u_1, \dots u_{i+1} \rangle) = 1$$

Por HI $\mathcal{L}' \in \mathbf{\Pi_i^p} \subseteq \mathbf{P}$. Luego existe la máquina determinística M' tal que $\mathcal{L}(M') = \mathcal{L}'$ y M' corre en tiempo polinomial.

$$\begin{array}{rcl} \langle x, u_1 \rangle \in \mathcal{L}' & \text{sii} & M'(\langle x, u_1 \rangle) = 1 \\ x \in \mathcal{L} & \text{sii} & \exists u_1 \in \{0,1\}^{q(|x|)}\ M'(\langle x, u_1 \rangle) = 1 \end{array}$$

Demostración de $\mathbf{P} = \mathbf{NP} \Rightarrow \mathbf{PH} = \mathbf{P}$.

Sup. $\mathbf{P} = \mathbf{NP}$. Probamos $\boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}, \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}} \subseteq \mathbf{P}$ por inducción en $i \geq 1$.
$i = 1$: trivial porque $\mathbf{P} = \mathbf{NP} = \boldsymbol{\Sigma}_{\mathbf{1}}^{\mathbf{p}}$ y $\mathbf{P} = \mathbf{coNP} = \boldsymbol{\Pi}_{\mathbf{1}}^{\mathbf{p}}$.
Sup. $\boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}, \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}} \subseteq \mathbf{P}$ y probemos $\boldsymbol{\Sigma}_{\boldsymbol{i+1}}^{\mathbf{p}}, \boldsymbol{\Pi}_{\boldsymbol{i+1}}^{\mathbf{p}} \subseteq \mathbf{P}$. Sea $\mathcal{L} \in \boldsymbol{\Sigma}_{\boldsymbol{i+1}}^{\mathbf{p}}$.
Existe una máquina determinística M que corre en tiempo polinomial y un polinomio q tal que

$$x \in \mathcal{L} \quad \text{sii} \quad \begin{array}{l} \exists u_1 \in \{0,1\}^{q(|x|)}\ \forall u_2 \in \{0,1\}^{q(|x|)} \dots \\ \cdots Q_{i+1} u_{i+1} \in \{0,1\}^{q(|x|)}\ M(\langle x, u_1, \dots u_{i+1} \rangle) = 1 \end{array}$$

Definimos $\mathcal{L}'$ así: $\langle x, u_1 \rangle \in \mathcal{L}'$ sii

$$\forall u_2 \in \{0,1\}^{q(|x|)} \cdots Q_{i+1} u_{i+1} \in \{0,1\}^{q(|x|)}\ M(\langle x, u_1, \dots u_{i+1} \rangle) = 1$$

Por HI $\mathcal{L}' \in \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}} \subseteq \mathbf{P}$. Luego existe la máquina determinística M' tal que $\mathcal{L}(M') = \mathcal{L}'$ y M' corre en tiempo polinomial.

$$\begin{array}{rcl} \langle x, u_1 \rangle \in \mathcal{L}' & \text{sii} & M'(\langle x, u_1 \rangle) = 1 \\ x \in \mathcal{L} & \text{sii} & \exists u_1 \in \{0,1\}^{q(|x|)}\ M'(\langle x, u_1 \rangle) = 1 \end{array}$$

Luego $\mathcal{L} \in \mathbf{NP} = \mathbf{P}$. Como $\mathcal{L}$ era arbitrario en $\boldsymbol{\Sigma}_{\boldsymbol{i+1}}^{\mathbf{p}}$, concluimos $\boldsymbol{\Sigma}_{\boldsymbol{i+1}}^{\mathbf{p}} \subseteq \mathbf{P}$ (y también $\boldsymbol{\Pi}_{\boldsymbol{i+1}}^{\mathbf{p}} \subseteq \mathbf{P}$). $\square$

Proposición

Si $\mathcal{L} \in \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$ y $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ entonces $\mathcal{L}' \in \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$.
Si $\mathcal{L} \in \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}}$ y $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ entonces $\mathcal{L}' \in \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}}$.

Proposición

Si $\mathcal{L} \in \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$ y $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ entonces $\mathcal{L}' \in \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$.
Si $\mathcal{L} \in \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}}$ y $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ entonces $\mathcal{L}' \in \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}}$.

Demostración.

Supongamos una máquina determinística M que corre en tiempo polinomial y un polinomio q tal que para todo x

$$x \in \mathcal{L} \text{ sii } \exists u_1 \in \{0,1\}^{q(|x|)} \ldots Q_i u_i \in \{0,1\}^{q(|x|)} \; M(\langle x, u_1, \ldots u_i \rangle) = 1.$$

Proposición

Si $\mathcal{L} \in \mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$ y $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ entonces $\mathcal{L}' \in \mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$.
Si $\mathcal{L} \in \mathbf{\Pi}_{\boldsymbol{i}}^{\mathbf{p}}$ y $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ entonces $\mathcal{L}' \in \mathbf{\Pi}_{\boldsymbol{i}}^{\mathbf{p}}$.

Demostración.

Supongamos una máquina determinística M que corre en tiempo polinomial y un polinomio q tal que para todo x

$$x \in \mathcal{L} \text{ sii } \exists u_1 \in \{0,1\}^{q(|x|)} \ldots Q_i u_i \in \{0,1\}^{q(|x|)} \; M(\langle x, u_1, \ldots u_i \rangle) = 1.$$

Supongamos que f es computable en tiempo polinomial y para todo x

$$x \in \mathcal{L}' \qquad \text{sii} \qquad f(x) \in \mathcal{L}$$

Proposición

Si $\mathcal{L} \in \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$ y $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ entonces $\mathcal{L}' \in \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$.
Si $\mathcal{L} \in \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}}$ y $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ entonces $\mathcal{L}' \in \boldsymbol{\Pi}_{\boldsymbol{i}}^{\mathbf{p}}$.

Demostración.

Supongamos una máquina determinística M que corre en tiempo polinomial y un polinomio q tal que para todo x

$$x \in \mathcal{L} \text{ sii } \exists u_1 \in \{0,1\}^{q(|x|)} \dots Q_i u_i \in \{0,1\}^{q(|x|)}\ M(\langle x, u_1, \dots u_i \rangle) = 1.$$

Supongamos que f es computable en tiempo polinomial y para todo x

$$x \in \mathcal{L}' \qquad \text{sii} \qquad f(x) \in \mathcal{L}$$

Juntando las dos, tenemos $x \in \mathcal{L}'$ sii

$$\exists u_1 \in \{0,1\}^{q(|x|)} \dots Q_i u_i \in \{0,1\}^{q(|x|)}\ \underbrace{M(\langle f(x), u_1, \dots u_i \rangle)}_{M'(\langle x, u_1, \dots u_i \rangle)} = 1.$$

M' también corre en tiempo polinomial. $\square$

Problemas $\boldsymbol{\Sigma_i^{\mathbf{p}}}$-completos

Completitud

Clase de complejidad: $\mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$-hard,$\mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$-completo

$\mathcal{L}$ es $\mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$**-hard** si $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ para todo $\mathcal{L}' \in \mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$.
$\mathcal{L}$ es $\mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$**-completo** si $\mathcal{L} \in \mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$ y $\mathcal{L} \in \mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$**-hard**

Análogamente definimos $\mathbf{\Pi}_{\boldsymbol{i}}^{\mathbf{p}}$**-hard**, $\mathbf{\Pi}_{\boldsymbol{i}}^{\mathbf{p}}$**-completo**, **PH-hard**, **PH-completo**.

¿Problemas **PH-completos**?

Proposición

Si existe $\mathcal{L} \in$ **PH-completo** entonces existe i tal que $\mathbf{PH} = \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$.

¿Problemas **PH-completos**?

Proposición

Si existe $\mathcal{L} \in$ **PH-completo** entonces existe i tal que $\mathbf{PH} = \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$.

Demostración.

Sea $\mathcal{L} \in \mathbf{PH} = \bigcup_i \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$ tal que $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ para todo $\mathcal{L}' \in \mathbf{PH}$.

¿Problemas **PH-completos**?

Proposición

Si existe $\mathcal{L} \in$ **PH-completo** entonces existe i tal que $\mathbf{PH} = \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$.

Demostración.

Sea $\mathcal{L} \in \mathbf{PH} = \bigcup_i \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$ tal que $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ para todo $\mathcal{L}' \in \mathbf{PH}$.
Sea i tal que $\mathcal{L} \in \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$.

¿Problemas **PH-completos**?

Proposición

Si existe $\mathcal{L} \in$ **PH-completo** entonces existe i tal que $\mathbf{PH} = \boldsymbol{\Sigma}_i^{\mathbf{p}}$.

Demostración.

Sea $\mathcal{L} \in \mathbf{PH} = \bigcup_i \boldsymbol{\Sigma}_i^{\mathbf{p}}$ tal que $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ para todo $\mathcal{L}' \in \mathbf{PH}$.
Sea i tal que $\mathcal{L} \in \boldsymbol{\Sigma}_i^{\mathbf{p}}$.
Sea $\mathcal{L}' \in \mathbf{PH}$ arbitrario. Como $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$, concluimos $\mathcal{L}' \in \boldsymbol{\Sigma}_i^{\mathbf{p}}$.
Luego $\mathbf{PH} \subseteq \boldsymbol{\Sigma}_i^{\mathbf{p}}$ (y $\mathbf{PH} \supseteq \boldsymbol{\Sigma}_i^{\mathbf{p}}$ es trivial). $\square$

¿Problemas **PH-completos**?

Proposición

Si existe $\mathcal{L} \in$ **PH-completo** entonces existe i tal que $\mathbf{PH} = \mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$.

Demostración.

Sea $\mathcal{L} \in \mathbf{PH} = \bigcup_i \mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$ tal que $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$ para todo $\mathcal{L}' \in \mathbf{PH}$.
Sea i tal que $\mathcal{L} \in \mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$.
Sea $\mathcal{L}' \in \mathbf{PH}$ arbitrario. Como $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$, concluimos $\mathcal{L}' \in \mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$.
Luego $\mathbf{PH} \subseteq \mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$ (y $\mathbf{PH} \supseteq \mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}}$ es trivial). $\square$

Se cree que para todo i, $\mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}} \subsetneq \mathbf{\Sigma}_{\boldsymbol{i+1}}^{\mathbf{p}}$. Se dice que la jerarquía polinomial **colapsa al i-ésimo nivel** si $\mathbf{\Sigma}_{\boldsymbol{i}}^{\mathbf{p}} = \mathbf{\Sigma}_{\boldsymbol{i+1}}^{\mathbf{p}}$ y en este caso, $\mathbf{\Sigma}_{\boldsymbol{p}}^{\mathbf{p}} = \mathbf{PH}$.
Como se cree que la jerarquía polinomial no colapsa al i-ésimo nivel, se cree que no existen los problemas **PH-completos**.

Ejercicio

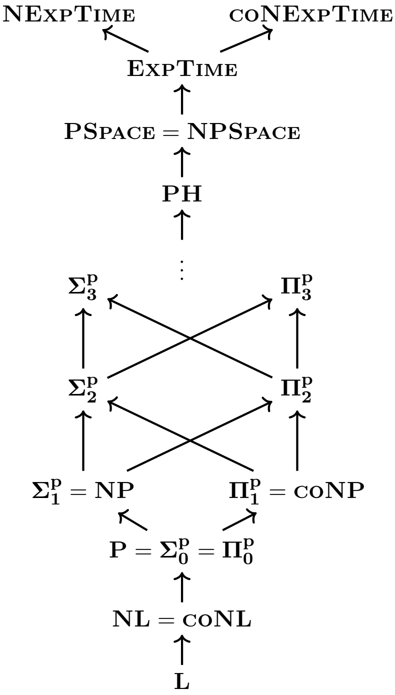
$\mathbf{PH} \subseteq \mathbf{PSPACE}$.

Corolario

Si la jerarquía polinomial no colapsa en ningún nivel, $\mathbf{PH} \neq \mathbf{PSPACE}$.

Demostración.

Si $\mathbf{PH} = \mathbf{PSPACE}$ entonces existirían problemas en **PH-completo** $=$ **PSPACE-completo**. $\square$


NExpTime
coNExpTime
ExpTime
PSpace = NPSpace
PH
Σ_3^p
Π_3^p
Σ_2^p
Π_2^p
Σ_1^p = NP
Π_1^p = coNP
P = Σ_0^p = Π_0^p
NL = coNL
L

Problemas $\mathbf{\Sigma}_i^{\mathbf{p}}$-completos

Problema: Satisfacibilidad de QBF acotada

$\Sigma_i\mathsf{SAT} = \{\langle\varphi\rangle :$ φ es una QBF de la forma $\exists\bar{y}_1\forall\bar{y}_2\ldots Q_i\bar{y}_i\ \psi(\bar{y}_1,\ldots,\bar{y}_i)$ donde las $\bar{y}_i$ son tuplas de variables booleanas, ψ es una fórmula booleana, los cuantificadores se alternan y $\models\varphi$ $\}$

Proposición

Para todo $i > 0$, $\Sigma_i\mathsf{SAT} \in \mathbf{\Sigma}_i^{\mathbf{p}}$**-completo**.

Demostración de $\Sigma_i\mathsf{SAT} \in \mathbf{\Sigma_i^p}$.

Consideremos la máquina determinística M que con entrada $\langle \varphi, u_1, \ldots, u_i \rangle$ hace esto :

> *si φ no es una QBF de la forma $\exists \bar{y}_1 \forall \bar{y}_2 \ldots Q_i \bar{y}_i \; \psi(\bar{y}_1, \ldots, \bar{y}_i)$, o si para algún j, $|u_j| < |\bar{y}_j|$, rechazar. Si no, devolver 1 si $(u_1 \restriction |y_1|) \ldots (u_i \restriction |y_i|) \models \psi$ y 0 en caso contrario*

Es claro que M corre en tiempo polinomial y $\langle \varphi \rangle \in \Sigma_i\mathsf{SAT}$ sii

$$\exists \bar{y}_1 \in \{0,1\}^k \; \forall \bar{y}_2 \{0,1\}^k \ldots Q_i \bar{y}_i \in \{0,1\}^k \; M(\langle \varphi, u_1, \ldots, u_i \rangle)$$

- $k = |\langle \varphi \rangle|$ es suficientemente largo
- la cantidad de variables de φ es siempre menor que la longitud de la codificación de φ

$\square$

Demostración de Σ_iSAT $\in \boldsymbol{\Sigma_i^{\mathrm{p}}}$-hard

Recordemos SAT $\in$ **NP-completo**. Sea $\mathcal{L} \in$ **NP** tal que

$$x \in \mathcal{L} \qquad \text{sii} \qquad \exists u \in \{0,1\}^{p(|x|)}\ M(xu) = 1$$

donde p es un polinomio y M es una máquina determinística que corre en tiempo polinomial $t(n)$

Para $j = 1 \ldots 4$ definimos fórmulas booleanas

$$\underset{\text{codifica}}{\psi_j} \left(\underbrace{\bar{e}}_{x}, \underbrace{\bar{c}}_{u}, \underbrace{\bar{q}_0}_{z_0}, \ldots, \underbrace{\bar{q}_m}_{z_m} \right)$$

donde $z_0, \ldots, z_m$, $z_i \in \{0,1\}^k$ (k depende solo de M) es una secuencia de mini-configuraciones, $m = t(|xu|)$, y $\bar{e}, \bar{c}, \bar{q}_0, \ldots, \bar{q}_m$ son tuplas de variables de tamaño polinomial.

$\psi_1 =$ "la entrada empieza con x y $u \in \{0,1\}^*$"

$\psi_2 =$ "z_0 es la configuración inicial"

$\psi_3 =$ "z_j evoluciona en z_{j+1} para $j = 0, \ldots, m-1$"

$\psi_4 =$ "z_m es una configuración final de M aceptadora"

$$\psi_j(\underbrace{\bar{e}}_{x}, \underbrace{\bar{c}}_{u}, \underbrace{\bar{q}_0}_{z_0}, \ldots, \underbrace{\bar{q}_m}_{z_m})$$

codifica

$$\begin{array}{lll} M(xu) = 1 & \text{sii} & \tilde{u} \models \exists \bar{e}, \bar{q}_0, \ldots, \bar{q}_m \ \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4 \\ & \text{sii} & \tilde{u} \models \forall \bar{e}, \bar{q}_0, \ldots, \bar{q}_m \ (\psi_1 \wedge \psi_2 \wedge \psi_3) \to \psi_4 \end{array}$$

donde

- $|u| = q(|x|)$
- $\tilde{u} = u(0)u(0)u(1)u(1)\ldots u(|u|-1)u(|u|-1)$ es la codificación de u
- corresponde a la valuación para las variables de $\bar{c}$

Entonces

$$\begin{array}{lll} x \in \mathcal{L} & \text{sii} & \exists u \in \{0,1\}^{p(|x|)} \ M(xu) = 1 \\ & \text{sii} & \models \exists \bar{c}, \exists \bar{e}, \bar{q}_0, \ldots, \bar{q}_m \ \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4 \\ & \text{sii} & \models \exists \bar{c} \ \forall \bar{e}, \bar{q}_0, \ldots, \bar{q}_m \ (\psi_1 \wedge \psi_2 \wedge \psi_3) \to \psi_4 \end{array}$$

Supongamos que $\mathcal{L} \in \boldsymbol{\Sigma_i^p}$. Sea M la máquina determinística *oblivious*, sin cinta de salida y con única cinta de trabajo que corre en tiempo polinomial $t(n)$ y para todo x

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

Supongamos que $\mathcal{L} \in \boldsymbol{\Sigma_i^p}$. Sea M la máquina determinística *oblivious*, sin cinta de salida y con única cinta de trabajo que corre en tiempo polinomial $t(n)$ y para todo x

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

Como antes:

$$\underset{\text{codifica}}{\psi_j} (\underbrace{\bar{e}}_{x}, \underbrace{\bar{c}}_{u=u_1\dots u_i}, \underbrace{\bar{q}_0}_{z_0}, \dots, \underbrace{\bar{q}_m}_{z_m})$$

donde $\bar{c}$ es una tupla de variables booleanas de tamaño $2 \cdot i \cdot q(|x|)$

$\psi_1 =$ "la entrada empieza con x y $u \in \{0,1\}^*$"

$\psi_2 =$ "z_0 es la configuración inicial"

$\psi_3 =$ "z_j evoluciona en z_{j+1} para $j = 0, \dots, m-1$"

$\psi_4 =$ "z_m es una configuración final de M aceptadora"

(no especifica las variables $\bar{c}$ que corresponden a $u = u_1 \dots u_i$)

Supongamos que $\mathcal{L} \in \boldsymbol{\Sigma_i^p}$. Sea M la máquina determinística *oblivious*, sin cinta de salida y con única cinta de trabajo que corre en tiempo polinomial $t(n)$ y para todo x

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

Como antes:

$$\underset{\text{codifica}}{\psi_j}\,(\underbrace{\bar{e}}_{x}, \underbrace{\bar{c}}_{u=u_1\ldots u_i}, \underbrace{\bar{q}_0}_{z_0}, \ldots, \underbrace{\bar{q}_m}_{z_m})$$

donde $\bar{c}$ es una tupla de variables booleanas de tamaño $2 \cdot i \cdot q(|x|)$

$\psi_1 =$ "la entrada empieza con x y $u \in \{0,1\}^*$"
$\psi_2 =$ "z_0 es la configuración inicial"
$\psi_3 =$ "z_j evoluciona en z_{j+1} para $j = 0, \ldots, m-1$"
$\psi_4 =$ "z_m es una configuración final de M aceptadora"

(no especifica las variables $\bar{c}$ que corresponden a $u = u_1 \ldots u_i$)

$M(xu_1 \ldots u_i) = 1$

$$\text{sii} \quad \tilde{u}_1 \ldots \tilde{u}_i \models \exists \bar{e}, \bar{q}_0, \ldots, \bar{q}_m \ \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

$$\text{sii} \quad \tilde{u}_1 \ldots \tilde{u}_i \models \forall \bar{e}, \bar{q}_0, \ldots, \bar{q}_m \ (\psi_1 \wedge \psi_2 \wedge \psi_3) \rightarrow \psi_4$$

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

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

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

  - ρ_x es una QBF con todas las tuplas cuantificadas de tamaño polinomial en $|x|$
  - ρ_x se calcula en tiempo polinomial a partir de x
  - cada $\bar{c}_i$ es una tupla de variables booleanas de dimensión $q(|x|)$ y $\bar{c} = \bar{c}_1, \ldots, \bar{c}_i$
  - el último bloque se convierte en un $\exists$, entonces ρ_x sigue teniendo $i-1$ alternancias de cuantificadores.

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

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

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

- El último bloque se convierte en un $\forall$, entonces ρ_x sigue teniendo $i-1$ alternancias de cuantificadores. $\square$