Complejidad Computacional

Santiago Figueira

Departamento de Computación - FCEN - UBA

clase 8

Clase 8

- Espacio logarítmico: **L** y **NL**
- Teorema de Immerman-Szelepcsényi

Espacio logarítmico: **L** y **NL**

Clase 8

- Espacio logarítmico: **L** y **NL**
- Teorema de Immerman-Szelepcsényi

L y NL

Clase de complejidad: L, NL

$$\mathbf{L} = \textbf{S}\textsc{pace}(\log n)$$

$$\mathbf{NL} = \textbf{NS}\textsc{pace}(\log n)$$

Observación

$\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P}$.

Demostración.

Si S es construible en espacio, sabemos que $\textbf{NS}\textsc{pace}(S(n)) \subseteq \textbf{DT}\textsc{ime}(2^{O(S(n))})$. Tomar $S(n) = \log n$. $\square$

L y NL

Clase de complejidad: L, NL

$$\mathbf{L} = \textbf{S}\textsc{pace}(\log n)$$

$$\mathbf{NL} = \textbf{NS}\textsc{pace}(\log n)$$

Observación

$\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P}$.

Demostración.

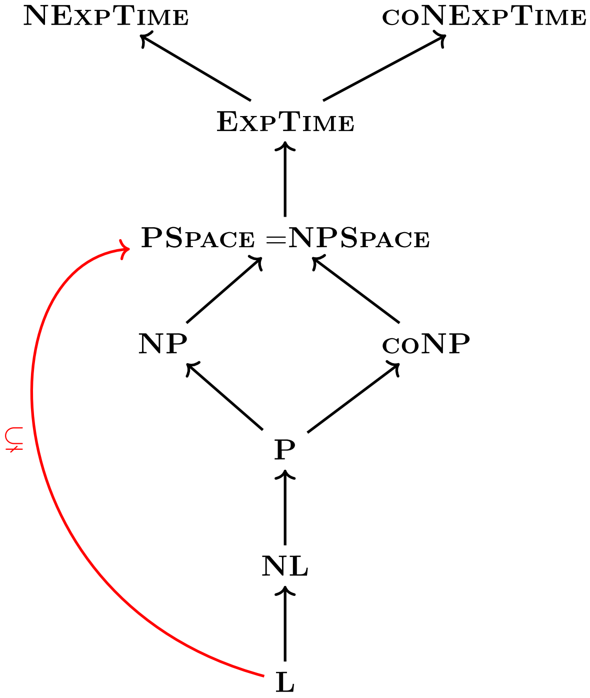
Si S es construible en espacio, sabemos que $\textbf{NS}\textsc{pace}(S(n)) \subseteq \textbf{DT}\textsc{ime}(2^{O(S(n))})$. Tomar $S(n) = \log n$. $\square$

Problema: EVEN (cantidad par de 1s)

$$\mathsf{EVEN} = \{x \in \{0,1\}^* : \text{hay una cantidad par de 1s en } x\}$$

Ejercicio

$\mathsf{EVEN} \in \mathbf{L}$.


NExpTime
coNExpTime
ExpTime
PSpace =NPSpace
NP
coNP
⊊
P
NL
L

Ejemplo de problema en $\mathbf{NL}$

Problema: PATH (existencia de camino)

$\mathsf{PATH} = \{\langle G, s, t\rangle : \text{hay un camino de } s \text{ a } t \text{ en el grafo dirigido } G\}$

No se sabe si $\mathsf{PATH} \in \mathbf{L}$, pero

Proposición

$\mathsf{PATH} \in \mathbf{NL}$.

Demostración.

Suponemos que los nodos de $G = \langle V, E \rangle$ están codificados como números del 0 a $|V| - 1$.

Definimos la máquina no-determinística N que con entrada $x = \langle G, s, t \rangle$ inventa un camino de s a t:

> $y \leftarrow s$; $m \leftarrow 0$
> *mientras* $m < |V|$:
> $z \leftarrow$ *inventar un valor en* $\{0, \ldots, |V| - 1\}$
>   *si* $(y, z) \notin E$ *(para esto, revisa* G*), 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}

Las variables y, z, m son posiciones contiguas de celdas en la cinta de trabajo. Solo almacenan números $\leq |V|$, de modo que alcanzan $\log |V|$ celdas para cada variable. Entonces M usa espacio $O(\log n)$.

$$N \text{ acepta } x \quad \text{sii} \quad x \in \mathsf{PATH},$$

luego $\mathsf{PATH} \in \mathbf{NL}$. $\square$

Reducibilidad para NL

Para la pregunta $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ usamos la (Karp) reducibilidad polinomial $\leq_{\mathrm{p}}$.

Para la pregunta $\mathbf{L} \stackrel{?}{=} \mathbf{NL}$ no nos sirve $\leq_{\mathrm{p}}$:

Proposición

Si $\mathcal{L} \notin \{\{0,1\}^*, \emptyset\}$ entonces $\mathcal{L}$ es **NL-hard** con respecto a $\leq_{\mathrm{p}}$.

Demostración.

Sea $a \in \mathcal{L}$ y $b \notin \mathcal{L}$. Tomemos $\mathcal{L}' \in \mathbf{NL}$. Sea f la función

$$f(x) = \begin{cases} a & \text{si } x \in \mathcal{L}' \\ b & \text{si } x \notin \mathcal{L}' \end{cases}$$

Como $\mathbf{NL} \subseteq \mathbf{P}$, f es computable en tiempo polinomial. Además,

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

Entonces $\mathcal{L}' \leq_{\mathrm{p}} \mathcal{L}$. $\square$

Funciones computables implícitamente en $\mathbf{L}$

Definición

Una función f es computable **implícitamente en L** si

- existe un polinomio p tal que para todo $x \in \{0,1\}^*$, $|f(x)| \leq p(x)$
- $\{\langle x, i\rangle \colon f(x)(i) = 1\}$ y $\{\langle x, i\rangle \colon i \leq |f(x)|\}$ están en $\mathbf{L}$.

La función booleana

$$\langle x, i\rangle \mapsto \begin{cases} f(x)(i) & \text{si } i \leq |f(x)| \\ 0 & \text{si no} \end{cases}$$

es $\mathbf{L}$

(Dado $\langle x, i\rangle$, decidir si $i \leq |f(x)|$ también es $\mathbf{L}$.)

Funciones computables en $\mathbf{L}$ sin contar salida

Definición

Una función f es **trabajo-L** computable si existe una máquina determinística que computa f en espacio $O(\log n)$ pero donde solo se cuenta el espacio de las cintas de trabajo y no de la cinta de salida. Como siempre,

- la cinta de salida es de solo escritura
- la cabeza escribe y se mueve a la derecha

Ejercicio

Una función es computable implícitamente en $\mathbf{L}$ sii es trabajo-$\mathbf{L}$ computable.

Composición de funciones computables implícitamente en $\mathbf{L}$

Proposición

Sean f, g computables implícitamente en $\mathbf{L}$. Entonces $g \circ f$ es computable implícitamente en $\mathbf{L}$.

Demostración

Sean M_f y M_g máquinas determinísticas que corren en espacio $O(\log n)$ tales que $M_f(\langle x, i\rangle) = f(x)(i)$ y $M_g(\langle x, i\rangle) = g(x)(i)$. Definimos la máquina M que con entrada $\langle x, i\rangle$ hace esto:

> $j \leftarrow 1$ *(índice de la entrada ficticia* $M_f(\langle x, j\rangle)$*)*
> $k \leftarrow 1$ *(índice de la salida de* $M_f(\langle x, j\rangle)$*)*
> $b \leftarrow$ *resultado de simular* $M_f(\langle x, j\rangle)$
> *simular* M_g *con entrada* $M_f(x)$ *paso a paso pero con estos cambios:*
>   *engañamos a* M_g *con la entrada:*
>   *el símbolo leído de la entrada de* M_g *es* b
>   *sea* I *la instrucción de* M_g *a ejecutar*
>   *(depende de* b*, de las cintas de trabajo de* M_g *y del estado de* M_g*)*
>   *si* I *mueve la cabeza de entrada a* R/L*,*
>   *incrementa/decrementa* j *en uno*
> $b \leftarrow$ *resultado de simular* $M_f(\langle x, j\rangle)$
>   *si* I *mueve la cabeza de salida (solo puede a* R*),*
>   *incrementa* k *en uno*
>   *si* I *escribe* $y \in \{0, 1, \square\}$ *en la salida y* $k = i$*,*
>   *escribir* y *en la cinta de salida de* M

- Cada vez que M_g lee un bit de $f(x)$, M lo calcula
  - no tiene espacio para calcular $f(x)$ entero, porque $|f(x)| \leq |x|^c$ para todo x suficientemente largo, pero $|x|^c$ es demasiado grande
- Para simular $M_f(\langle x, j\rangle)$ necesita leer el x de la entrada original de M porque no tiene espacio para copiar x en otro lado
- Lleva cuenta de las cintas de trabajo de M_g
- $M(\langle x, i\rangle) = g(f(x))(i)$
- Para las variables j y k usa

$$\leq \log |f(x)| = \log(|x|^c) = O(\log |x|)$$

- El resto del espacio que usa es el de que usan M_g y M_f.

$\square$

Reducibilidad **L**

Definición

$\mathcal{L}$ es **L-reducible** a $\mathcal{L}'$, notado $\mathcal{L} \leq_\ell \mathcal{L}'$, si existe una función f computable implícitamente en **L** tal que para todo x,

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

En este caso decimos que $\mathcal{L} \leq_\ell \mathcal{L}'$ **vía** f.

Clase de complejidad: **NL-hard**, **NL-completo**

$\mathcal{L}$ es **NL-hard** si $\mathcal{L}' \leq_\ell \mathcal{L}$ para todo $\mathcal{L}' \in$ **NL**.
$\mathcal{L}$ es **NL-completo** si $\mathcal{L} \in$ **NL** y $\mathcal{L} \in$ **NL-hard**.

Reducibilidad $\mathbf{L}$

Ejercicio

Si $\mathcal{L} \leq_\ell \mathcal{L}'$ y $\mathcal{L}' \in \mathbf{L}$, entonces $\mathcal{L} \in \mathbf{L}$.

Ejercicio

La relación $\leq_\ell$ es transitiva.

Teorema de Immerman-Szelepcsényi

Clase 8

Espacio logarítmico: **L** y **NL**

Teorema de Immerman-Szelepcsényi

PATH $\in$ **NL-completo**

Teorema

PATH $\in$ **NL-completo**.

Demostración.

Ya vimos que PATH $\in$ **NL**.

Solo falta ver PATH $\in$ **NL-hard**. $\square$

PATH $\in$ **NL-completo**

Teorema

PATH $\in$ **NL-completo**.

Demostración.

Ya vimos que PATH $\in$ **NL**.

Solo falta ver PATH $\in$ **NL-hard**. $\square$

Corolario

$\overline{\text{PATH}}$ $\in$ **coNL-completo**.

Demostración de PATH $\in$ **NL-hard**.

Sea $\mathcal{L} \in \mathbf{NL}$ y N una máquina no-determinística tal que $\mathcal{L}(N) = \mathcal{L}$ y N usa espacio $O(\log n)$. Veamos que $\mathcal{L} \leq_\ell$ PATH vía f.

$$f(x) = \langle G_{N,x}, C_0, C_f \rangle$$

donde C_0 es la configuración inicial de N para x y C_f es la final.

Demostración de PATH $\in$ **NL-hard**.

Sea $\mathcal{L} \in \mathbf{NL}$ y N una máquina no-determinística tal que $\mathcal{L}(N) = \mathcal{L}$ y N usa espacio $O(\log n)$. Veamos que $\mathcal{L} \leq_\ell \mathsf{PATH}$ vía f.

$$f(x) = \langle G_{N,x}, C_0, C_f \rangle$$

donde C_0 es la configuración inicial de N para x y C_f es la final. Cada configuración C se codifica con $c \cdot \log |x|$ bits.

$$\begin{array}{lll} x \in \mathcal{L} & \text{sii} & N \text{ acepta } x \\ & \text{sii} & \text{existe un camino desde } C_0 \text{ hasta } C_f \text{ en } G_{N,x} \\ & \text{sii} & f(x) \in \mathsf{PATH} \end{array}$$

Demostración de PATH $\in$ **NL-hard**.

Sea $\mathcal{L} \in \mathbf{NL}$ y N una máquina no-determinística tal que $\mathcal{L}(N) = \mathcal{L}$ y N usa espacio $O(\log n)$. Veamos que $\mathcal{L} \leq_\ell \mathsf{PATH}$ vía f.

$$f(x) = \langle G_{N,x}, C_0, C_f \rangle$$

donde C_0 es la configuración inicial de N para x y C_f es la final. Cada configuración C se codifica con $c \cdot \log |x|$ bits.

$$\begin{array}{lll} x \in \mathcal{L} & \text{sii} & N \text{ acepta } x \\ & \text{sii} & \text{existe un camino desde } C_0 \text{ hasta } C_f \text{ en } G_{N,x} \\ & \text{sii} & f(x) \in \mathsf{PATH} \end{array}$$

Falta ver que f es computable implícitamente en $\mathbf{L}$.

- El grafo $G_{N,x}$ tiene $2^{c \cdot \log |x|}$ nodos
- Lo representamos con la matriz de adyacencia, de dimensión $2^{c \cdot \log |x|} \times 2^{c \cdot \log |x|}$.
- $|f(x)| = O((2^{c \cdot \log |x|})^2) = O(|x|^{2 \cdot c})$
- Dado $\langle x, i \rangle$, calculamos el i-ésimo bit de $f(x)$ usando espacio logarítmico (enumera configuraciones reusando el espacio).

Caracterización de **NL** con certificados

Teorema

$\mathcal{L} \in \mathbf{NL}$ sii existe un polinomio $p : \mathbb{N} \to \mathbb{N}$ y una máquina determinística M con una cinta adicional de lectura de una única vez (lee y pasa a la siguiente celda a la derecha; no puede volver atrás) tal que

- M usa espacio $O(\log n)$
  (como siempre, solo cuentan sus cintas de trabajo y salida)
- para todo x:

  $$x \in \mathcal{L} \quad \text{sii} \quad \text{existe } u \in \{0,1\}^{p(|x|)} \text{ tal que } M(x,u) = 1$$

  donde $M(x,u)$ denota la salida de M cuando
  - la cinta de entrada tiene x
  - la cinta adicional de lectura de una única vez tiene u
- la cinta de entrada y la cinta adicional no cuentan en el espacio usado por M

- M se llama **verificador**
- u se llama **certificado**

$\overline{\mathsf{PATH}} \in \mathbf{NL}$

Teorema (Immerman-Szelepcsényi)

$\overline{\mathsf{PATH}} \in \mathbf{NL}$

Corolario

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

Demostración.

Veamos $\mathbf{coNL} \subseteq \mathbf{NL}$ (el caso $\mathbf{NL} \subseteq \mathbf{coNL}$ es análogo). Sea $\mathcal{L} \in \mathbf{coNL}$. Como $\overline{\mathsf{PATH}} \in \mathbf{coNL}$**-completo**, tenemos $\mathcal{L} \leq_\ell \overline{\mathsf{PATH}} \in \mathbf{NL}$. Luego $\mathcal{L} \in \mathbf{NL}$. $\square$

Distinto a lo que pasa con $\mathbf{NP}$ vs $\mathbf{coNP}$.

- $\mathbf{NP} \stackrel{?}{=} \mathbf{coNP}$
- se cree que $\mathbf{NP} \neq \mathbf{coNP}$

Demostración de $\overline{\mathsf{PATH}} \in \mathbf{NL}$

Definimos un verificador M que

- usa espacio logarítmico; no cuenta cinta de entrada ni cinta de lectura de única vez donde va el certificado Z
- certifica que $x \notin \mathsf{PATH}$

$$x \in \overline{\mathsf{PATH}} \quad \text{sii} \quad \text{existe } Z \in \{0,1\}^{p(|x|)} \text{ tal que } M(x,Z) = 1$$

Demostración de $\overline{\mathsf{PATH}} \in \mathbf{NL}$

Definimos un verificador M que

- usa espacio logarítmico; no cuenta cinta de entrada ni cinta de lectura de única vez donde va el certificado Z
- certifica que $x \notin \mathsf{PATH}$

$$x \in \overline{\mathsf{PATH}} \quad \text{sii} \quad \text{existe } Z \in \{0,1\}^{p(|x|)} \text{ tal que } M(x,Z) = 1$$

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

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

Notar que A_n es la componente conexa de s en G. Luego

$$\langle G, s, t\rangle \notin \mathsf{PATH} \quad \text{sii} \quad t \notin A_n.$$

Vamos a dar varios certificados Z de distintos hechos.
Hay un verificador que puede revisar que Z sea un certificado válido para el hecho que pretende certificar

- solo puede leer Z de izquierda a derecha una celda a la vez sin volver atrás
- solo puede usar espacio logarítmico para su revisión
- puede buscar lo que quiera en la cinta de entrada (mover la cabeza a izquierda y derecha)
- puede guardar partes de la entrada o partes del certificado en cintas de trabajo, siempre que no excedan el espacio logarítmico

Siempre codificamos a los nodos de G en binario (tamaño $O(\log n)$).

Recordar que

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

Certificado de que $v \in A_i$: una lista de nodos

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

tal que

- cada v_i es la codificación en binario de un nodo de V
- $v_0 = s$
- $(v_j, v_{j+1}) \in E$
- $v_k = v$
- $k \leq i$

La lista que muestra $Z_{v \in A_i}$ es una forma de probar que $v \in A_i$.

Notar que el tamaño de $Z_{v \in A_i}$ es polinomial.

El verificador chequea el certificado en espacio $O(\log n)$.

Certificado de que $v \notin A_i$ *conociendo* $|A_i|$: una lista de pares

$$Z_{v \notin A_i}^{|A_i|} = \langle (v_1, Z_{v_1 \in A_i}), (v_2, Z_{v_2 \in A_i}), \ldots, (v_k, Z_{v_k \in A_i}) \rangle$$

tal que

- cada v_i es la codificación en binario de un nodo de V
- $k = |A_i|$
- $v_j < v_{j+1}$
- $v \notin \{v_1, \ldots, v_k\}$

$Z_{v \notin A_i}^{|A_i|}$ muestra k elementos distintos en A_i tal que ninguno es v y $|A_i| = k$. Es una forma de probar que $v \notin A_i$.

Notar que el tamaño de $Z_{v \notin A_i}^{|A_i|}$ es polinomial.

El verificador chequea el certificado en espacio $O(\log n)$.

Certificado de que $v \notin A_i$ *conociendo* $|A_{i-1}|$: una lista de pares

$$Z^{|A_{i-1}|}_{v \notin A_i} = \langle (v_1, Z_{v_1 \in A_{i-1}}), (v_2, Z_{v_2 \in A_{i-1}}), \ldots, (v_k, Z_{v_k \in A_{i-1}}) \rangle$$

tal que

- cada v_i es la codificación en binario de un nodo de V
- $k = |A_{i-1}|$
- $v_j < v_{j+1}$
- $v \notin \{v_1, \ldots, v_k\}$
- $v \notin \bigcup_{1 \leq j \leq k} E(v_i)$, donde $E(x) = \{y \in V : (x, y) \in E\}$

Notar que el tamaño de $Z^{|A_{i-1}|}_{v \notin A_i}$ es polinomial.

El verificador chequea el certificado en espacio $O(\log n)$.

Recordar que $V = \{1, \ldots, n\}$

Certificado de que $|A_i| = a$ *conociendo* $|A_{i-1}|$: una lista de pares

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

tal que

- si $v \in A_i$ entonces $Z_v = Z_{v \in A_i}$
- si $v \notin A_i$ entonces $Z_v = Z^{|A_i|-1}_{v \notin A_i}$
- $|v \colon Z_v = Z_{v \in A_i}| = |A_i| = a$

Notar que el tamaño de $Z^{|A_{i-1}|}_{|A_i|=a}$ es polinomial.

El verificador chequea el certificado en espacio $O(\log n)$.

Certificado final de que $t \notin A_n$:

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

tal que

- $a_i = |A_i|$ para todo i

Notar que el tamaño de Z es polinomial.

El verificador chequea el certificado en espacio $O(\log n)$:

- sabemos que $A_0 = \{s\}$ y por lo tanto $|A_0| = 1$
- a medida que lee de izquierda a derecha, va guardando $|A_{i-1}|$ para verificar $Z^{|A_{i-1}|}_{|A_i|=a_i}$ (reusa espacio)

$\square$

Teorema de Immerman-Szelepcsényi más general

Teorema

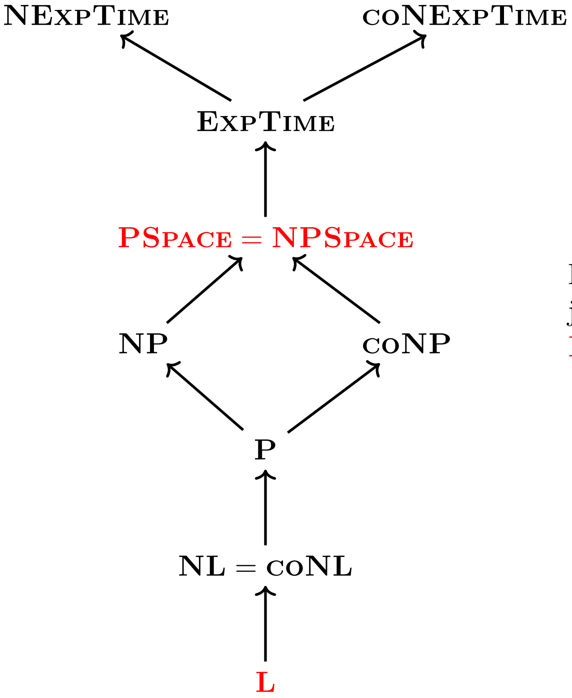
Si $S(n) \geq \log n$ es construible en espacio entonces $\mathbf{NS{\scriptstyle PACE}}(S(n)) = \mathbf{coNS{\scriptstyle PACE}}(S(n))$.

Ejercicio

Demostrarlo.

$\mathbf{NL} = \mathbf{coNL}$ es un caso particular cuando $S(n) = \log n$.

$$
\begin{tikzcd}
\textbf{N}\textsc{\textbf{ExpTime}} & & \textsc{\textbf{co}}\textbf{N}\textsc{\textbf{ExpTime}} \\
& \textsc{\textbf{ExpTime}} \arrow[ul] \arrow[ur] & \\
& \textbf{P}\textsc{\textbf{Space}} = \textbf{NP}\textsc{\textbf{Space}} \arrow[u] & \\
\textbf{NP} \arrow[ur] & & \textsc{\textbf{co}}\textbf{NP} \arrow[ul] \\
& \textbf{P} \arrow[ul] \arrow[ur] & \\
& \textbf{NL} = \textsc{\textbf{co}}\textbf{NL} \arrow[u] & \\
& \textbf{L} \arrow[u] &
\end{tikzcd}
$$



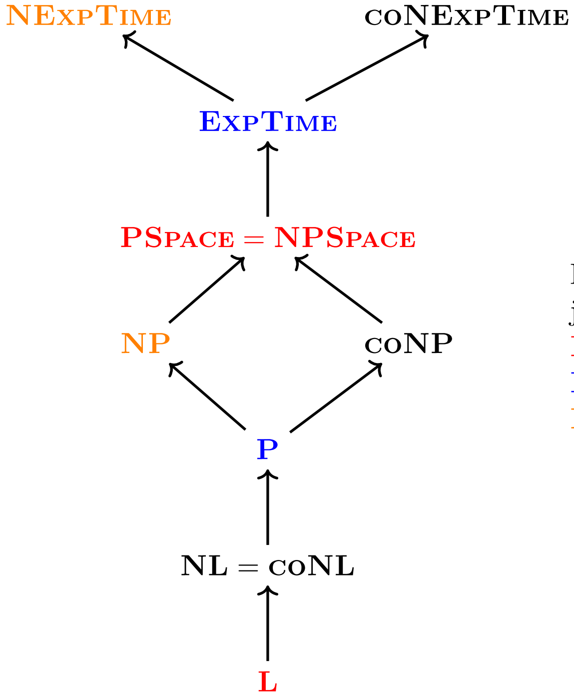
Por los teoremas de jerarquías:
$\mathbf{L} \subsetneq \mathbf{PSpace}$

$$\begin{tikzcd}
\mathbf{NExpTime} & & \mathbf{coNExpTime} \\
& \mathbf{ExpTime} \arrow[ul] \arrow[ur] & \\
& \mathbf{PSpace} = \mathbf{NPSpace} \arrow[u] & \\
\mathbf{NP} \arrow[ur] & & \mathbf{coNP} \arrow[ul] \\
& \mathbf{P} \arrow[ul] \arrow[ur] & \\
& \mathbf{NL} = \mathbf{coNL} \arrow[u] & \\
& \mathbf{L} \arrow[u] &
\end{tikzcd}$$

Por los teoremas de jerarquías:

$\mathbf{L} \subsetneq \mathbf{PSpace}$

$\mathbf{P} \subsetneq \mathbf{ExpTime}$



Por los teoremas de jerarquías:
$\mathbf{L} \subsetneq \mathbf{PSpace}$
$\mathbf{P} \subsetneq \mathbf{ExpTime}$
$\mathbf{NP} \subsetneq \mathbf{NExpTime}$