Clases de Complejidad

IIC3242

Clases de complejidad: Supuestos

Vamos a definir y estudiar las propiedades fundamentales de algunas de las clases de complejidad más importantes.

Modelo para este estudio: Máquina de Turing (determinista o no determinista) con un número arbitrario de cintas.

- Mostramos como se define el tiempo de ejecución para estas máquinas.
- También vimos como se define el espacio ocupado para las MTs con una cinta de lectura y otra de trabajo.
  - ¿Cómo se define esto para una MT con $k \geq 2$ cintas de trabajo?

Clases de complejidad: Supuestos

También es necesario imponer una restricción sobre las funciones usadas para medir la complejidad.

Supuesto

Cada función f para medir la complejidad es *edificable*, vale decir, existe una MT M determinista, con una cinta de entrada y $k \geq 1$ cintas de trabajo tal que:

- t_M es $O(n + f(n))$
- s_M es $O(f(n))$
- M acepta todas las entradas
- Si la entrada de M tiene largo n, entonces al detenerse M tiene en la k-ésima cinta de trabajo el símbolo $\vdash$, seguido de $f(n)$ símbolos $\#$, y estos seguidos sólo del símbolo B.

Clases de complejidad deterministas: Tiempo

Dado: Alfabeto Σ.

DTIME(t): conjunto de todos los lenguajes $L \subseteq \Sigma^*$ que pueden ser aceptados en tiempo $O(t)$ por una MT determinista.

Dos clases fundamentales:

$$\text{PTIME} = \bigcup_{k \in \mathbb{N}} \text{DTIME}(n^k)$$

$$\text{EXPTIME} = \bigcup_{k \in \mathbb{N}} \text{DTIME}(2^{n^k})$$

PTIME: Conjunto de todos los problemas que pueden ser solucionados "eficientemente".

Clases de complejidad no deterministas: Tiempo

NTIME(t): conjunto de todos los lenguajes $L \subseteq \Sigma^*$ que pueden ser aceptados en tiempo $O(t)$ por una MT no determinista.

Dos clases fundamentales:

$$\text{NP} = \bigcup_{k \in \mathbb{N}} \text{NTIME}(n^k)$$

$$\text{NEXPTIME} = \bigcup_{k \in \mathbb{N}} \text{NTIME}(2^{n^k})$$

Clases de complejidad deterministas: Espacio

DSPACE(s): conjunto de todos los lenguajes $L \subseteq \Sigma^*$ que pueden ser aceptados en espacio $O(s)$ por una MT determinista.

Tres clases fundamentales:

$$\begin{aligned} \text{LOGSPACE} &= \text{DSPACE}(\log n) \\ \text{PSPACE} &= \bigcup_{k \in \mathbb{N}} \text{DSPACE}(n^k) \\ \text{EXPSPACE} &= \bigcup_{k \in \mathbb{N}} \text{DSPACE}(2^{n^k}) \end{aligned}$$

Clases de complejidad no deterministas: Espacio

NSPACE(s): conjunto de todos los lenguajes $L \subseteq \Sigma^*$ que pueden ser aceptados en espacio $O(s)$ por una MT no determinista.

Tres clases fundamentales:

$$\begin{aligned} \text{NLOGSPACE} &= \text{NSPACE}(\log n) \\ \text{NPSPACE} &= \bigcup_{k \in \mathbb{N}} \text{NSPACE}(n^k) \\ \text{NEXPSPACE} &= \bigcup_{k \in \mathbb{N}} \text{NSPACE}(2^{n^k}) \end{aligned}$$

Clases de complejidad: Complemento

Dado un lenguaje L sobre un alfabeto Σ:

$$\overline{L} = \{w \in \Sigma^* \mid w \notin L\}.$$

Definición

Dada una clase de complejidad $\mathcal{C}$, el conjunto de los complementos de $\mathcal{C}$ se define como: co-$\mathcal{C} = \{\overline{L} \mid L \in \mathcal{C}\}$

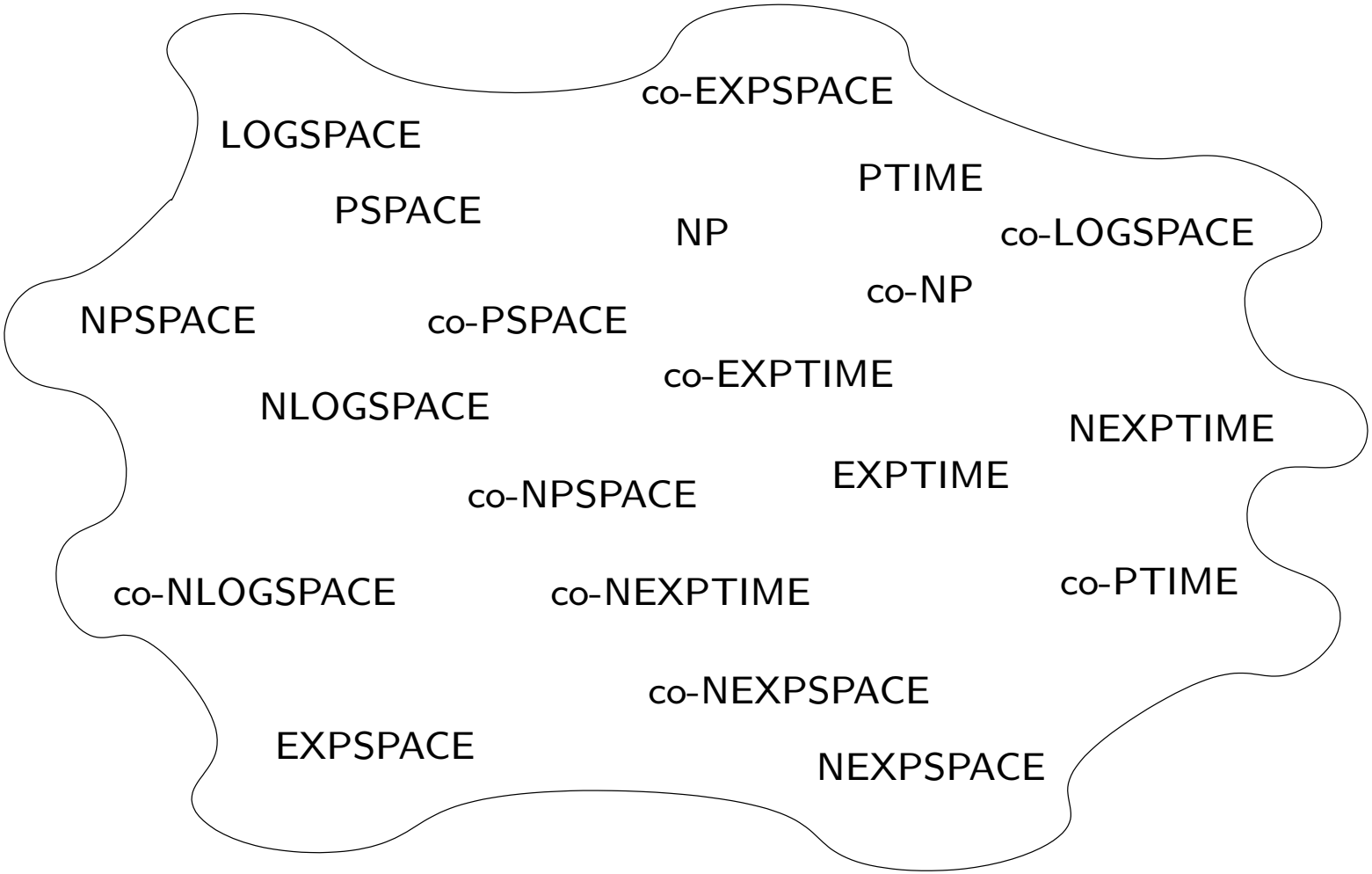
Ejemplo

Una clase muy estudiada: co-NP

- ¿Puede identificar algún problema en esta clase? ¿Es esta clase igual a NP?

Clases de complejidad: Relaciones

Punto de partida:



Resultados básicos

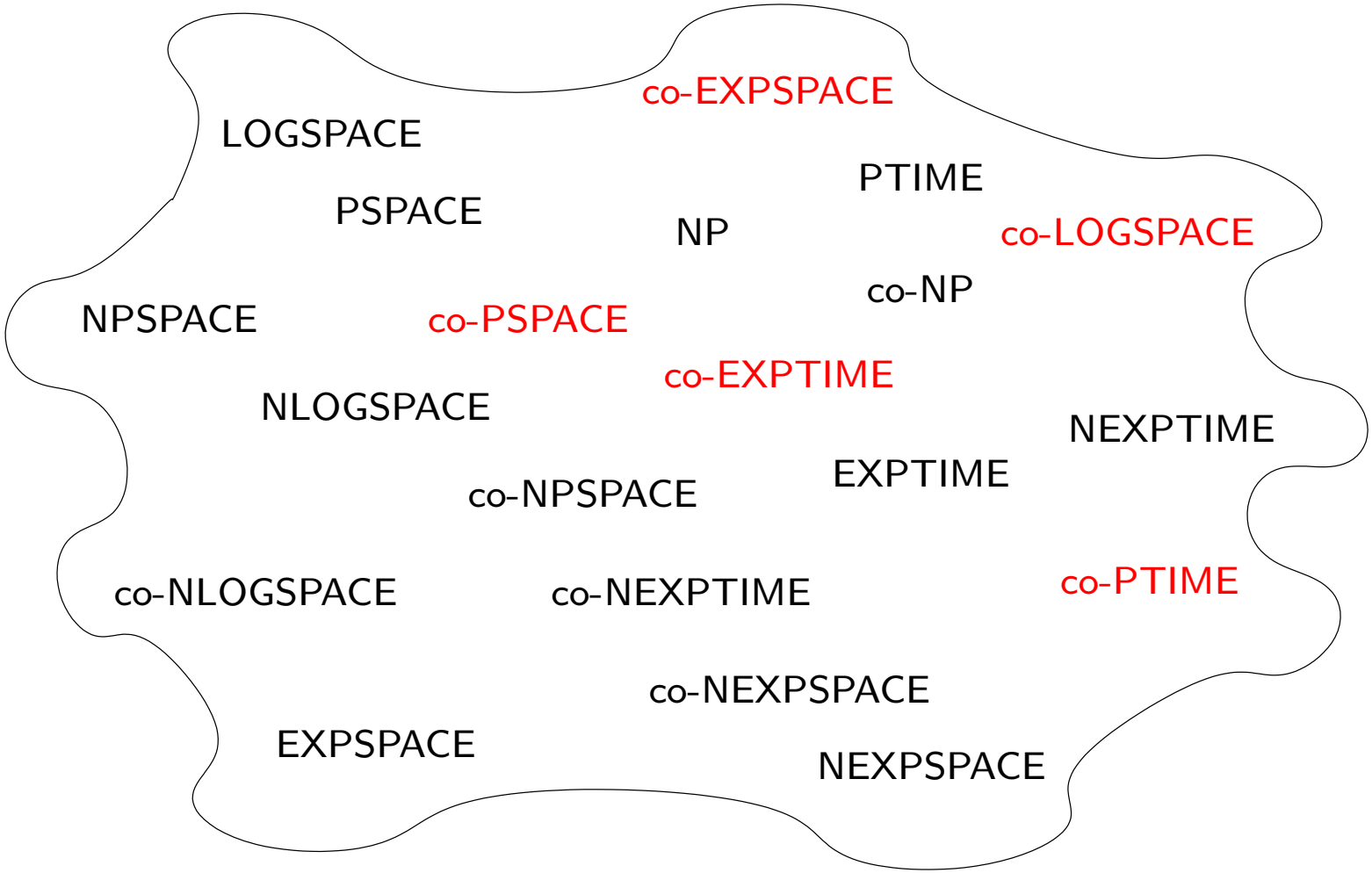
Las siguientes propiedades son triviales:

- Si $\mathcal{C}$ es una clase de complejidad definida en términos de MTs deterministas, entonces co-$\mathcal{C} = \mathcal{C}$.
- Si f es $O(g)$, entonces:
  - $\text{DTIME}(f) \subseteq \text{DTIME}(g)$
  - $\text{NTIME}(f) \subseteq \text{NTIME}(g)$
  - $\text{DSPACE}(f) \subseteq \text{DSPACE}(g)$
  - $\text{NSPACE}(f) \subseteq \text{NSPACE}(g)$
- $\text{DTIME}(f) \subseteq \text{NTIME}(f)$
- $\text{DSPACE}(f) \subseteq \text{NSPACE}(f)$

La figura empieza a tomar forma ...

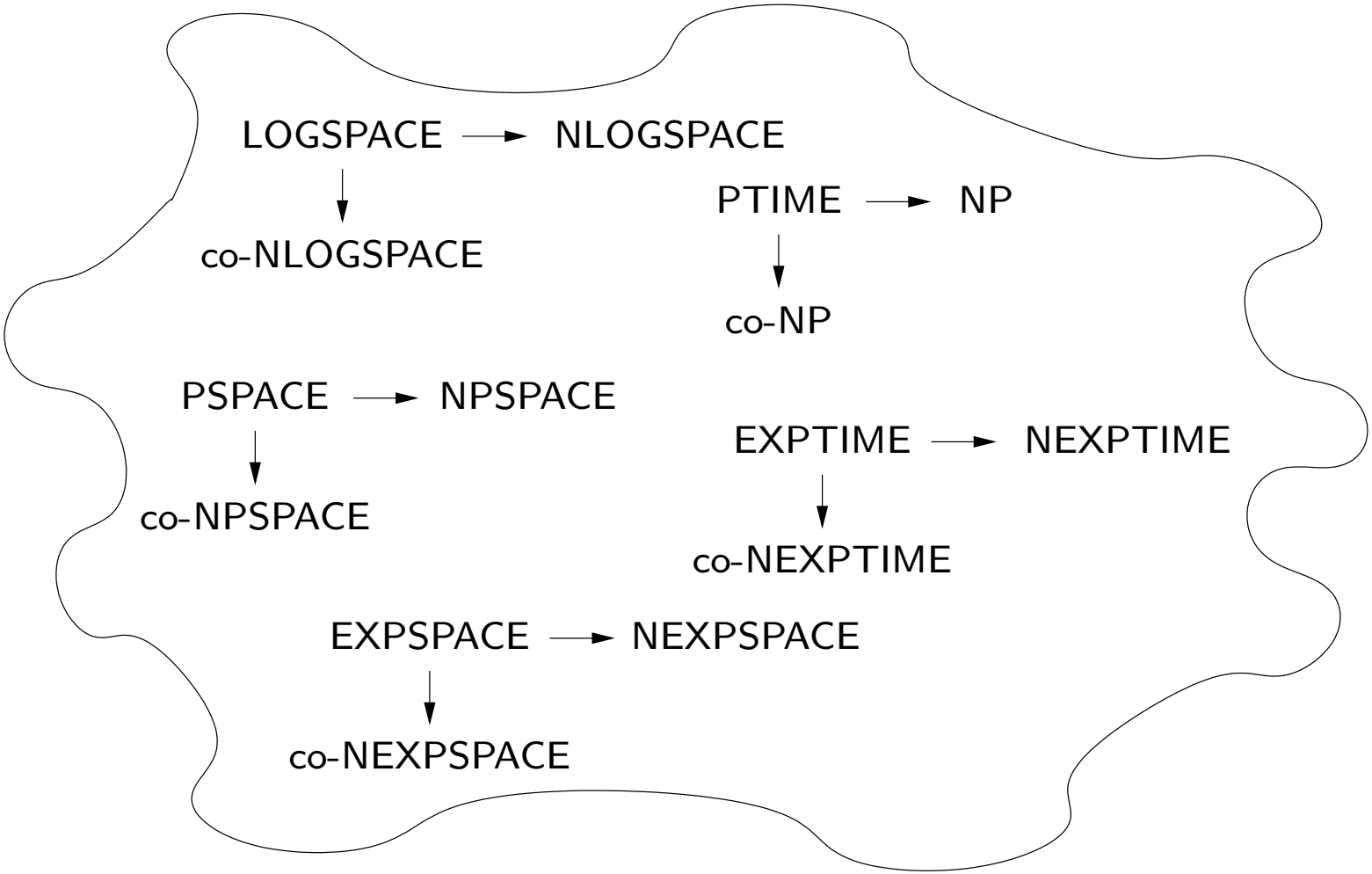
Clases de complejidad: Relaciones

Podemos eliminar algunas clases:



Clases de complejidad: Relaciones

También tenemos algunas relaciones de inclusión:



Relación entre tiempo y espacio

Teorema

$$NTIME(f(n)) \subseteq DSPACE(f(n))$$

Ejercicio

Demuestre el teorema.

Corolario

$$\begin{aligned} NP &\subseteq PSPACE \\ co\text{-}NP &\subseteq PSPACE \\ NEXPTIME &\subseteq EXPSPACE \\ co\text{-}NEXPTIME &\subseteq EXPSPACE \end{aligned}$$

Relación entre tiempo y espacio

Teorema

Si $f(n)$ *es* $\Omega(\log n)$, *entonces:*

$$NSPACE(f(n)) \subseteq \bigcup_{k \in \mathbb{N}} DTIME(2^{k \cdot f(n)})$$

Ejercicio

Demuestre el teorema.

- ¿Por qué se necesita que $f(n)$ sea $\Omega(\log n)$ para demostrar el teorema?

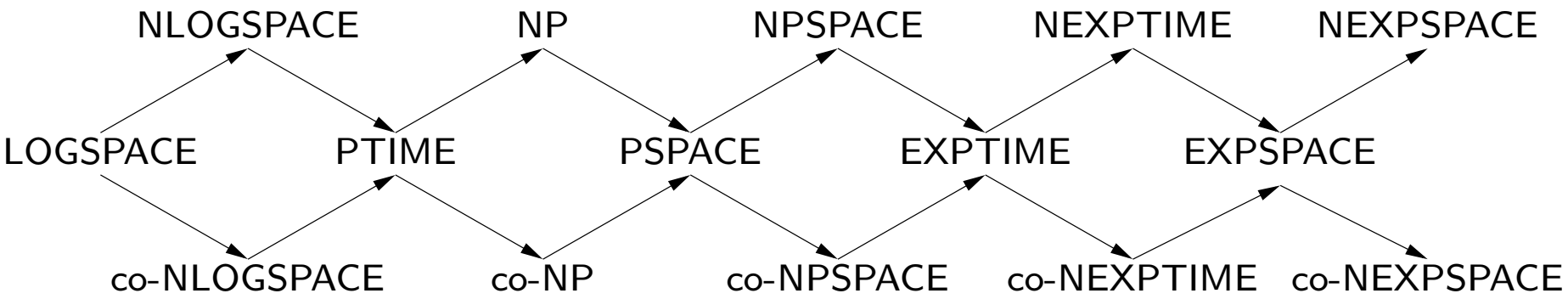
Relación entre tiempo y espacio

Corolario

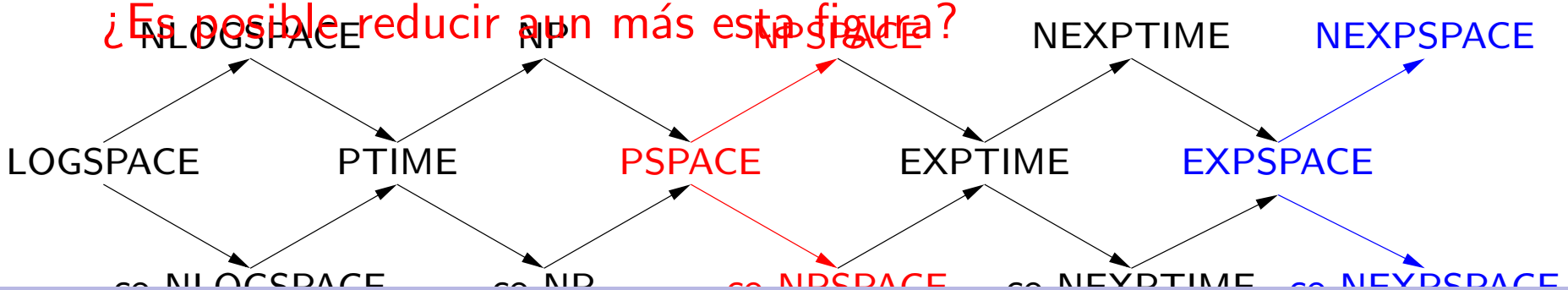
$$
\begin{aligned}
\mathit{NLOGSPACE} &\subseteq \mathit{PTIME} \\
\mathit{co\text{-}NLOGSPACE} &\subseteq \mathit{PTIME} \\
\mathit{NPSPACE} &\subseteq \mathit{EXPTIME} \\
\mathit{co\text{-}NPSPACE} &\subseteq \mathit{EXPTIME}
\end{aligned}
$$

Clases de complejidad: Relaciones

Como corolario de los resultados anteriores obtenemos:



¿Es posible reducir aun más esta figura?



Teorema de Savitch

Teorema (Savitch)

Si $f(n)$ *es* $\Omega(\log n)$, *entonces* $NSPACE(f(n)) \subseteq DSPACE(f(n)^2)$.

Demostración: Sea $L \in \text{NSPACE}(f(n))$. Entonces existe una MT no determinista $M = (Q, \Sigma, \Gamma, q_0, \delta, F)$ tal que:

- M tiene una cinta de lectura y $k \geq 1$ cintas de escritura
- $L = L(M)$
- s_M es $O(f(n))$

Teorema de Savitch: Demostración

Con una entrada de largo n, la cantidad de configuraciones de M está acotado por:

$$|Q| \cdot (n+2) \cdot s_M(n)^k \cdot |\Gamma|^{k \cdot s_M(n)}$$

Como s_M es $O(f(n))$ y $f(n)$ es $\Omega(\log n)$, existe una constante c tal que la cantidad de configuraciones de M está acotada por:

$$2^{c \cdot f(n)}$$

Además, existe una constante d tal que el espacio necesario para almacenar una configuración de M está acotado por $d \cdot f(n)$.

- ¿Por qué?

Teorema de Savitch: Demostración

Considere la siguiente función:

verificar(α: configuración, β: configuración, k: número natural)
  if $\alpha = \beta$ **then return**(true)
  else if $k \geq 1$ y β es sucesor de α en M **then return**(true)
  else if $k \geq 2$ **then**
    for each configuración γ de M **do**
      if verificar(α, γ, $\lfloor \frac{k}{2} \rfloor$) y **verificar**($\gamma$, β, $\lceil \frac{k}{2} \rceil$) **then return**(true)
  return(false)

Teorema de Savitch: Demostración

Sea α_0 la configuración inicial de M con entrada w.

Para verificar si $w \in L$, basta verificar si existe alguna configuración final β de M tal que **verificar**$(\alpha_0, \beta, 2^{c \cdot f(n)})$ retorna true.

Esto puede implementarse en espacio $O(f(n)^2)$ en una MT determinista.

- ¿Cómo puede hacerse esto?

Concluimos que $L \in \text{DSPACE}(f(n)^2)$.

□

Teorema de Savitch: Algunas consecuencias

Del Teorema de Savitch obtenemos los siguientes corolarios:

Corolario

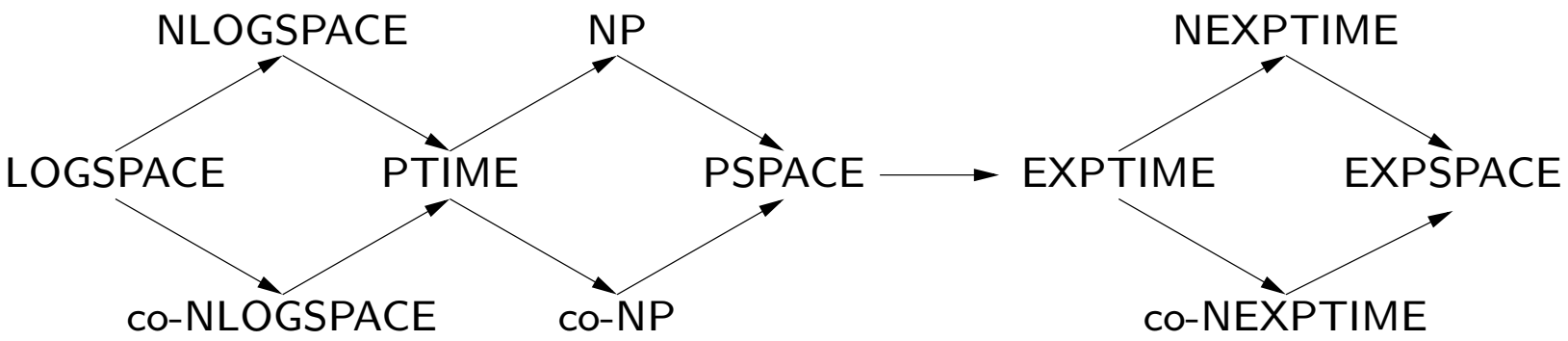
$$\begin{aligned} PSPACE &= NPSPACE \\ EXPSPACE &= NEXPSPACE \end{aligned}$$

Combinando esto con los resultados anteriores, obtenemos que:

- PSPACE = NPSPACE = co-NPSPACE
- EXPSPACE = NEXPSPACE = co-NEXPSPACE

Clases de complejidad: Relaciones y el Teorema de Savitch

Tenemos un mejor entendimiento de la relación entre las clases de complejidad:



Y ahora seguimos con ¿NLOGSPACE = co-NLOGSPACE?

- La respuesta a esta pregunta **no** se deduce del Teorema de Savitch.

Clases de complejidad: Teorema de Immerman-Szelepcsényi

Teorema (Immerman-Szelepcsényi)

Si $f(n)$ *es* $\Omega(\log n)$, *entonces* $NSPACE(f(n)) = co\text{-}NSPACE(f(n))$.

Demostración: Sea $L \in \text{NSPACE}(f(n))$. Entonces existe una MT no determinista $M = (Q, \Sigma, \Gamma, q_0, \delta, F)$ tal que:

- M tiene una cinta de lectura y $k \geq 1$ cintas de escritura
- $L = L(M)$
- s_M es $O(f(n))$

Teorema de Immerman-Szelepcsényi: Demostración

Suponemos que cuando M va a aceptar una palabra w:

- borra el contenido de las k cintas de escritura
- dejas las $k+1$ cabezas lectoras en la posición 1
- para en un único estado final

Podemos suponer esto sin perdida de generalidad.

- ¿Por qué?

Ventaja de esta suposición: Existe una única configuración de aceptación

Teorema de Immerman-Szelepcsényi: Demostración

Sea w una entrada de M de largo n

Como en la demostración del Teorema de Savitch:

- Existe una constante c tal que la cantidad de configuraciones de M está acotada por:

$$2^{c \cdot f(n)}$$

- Existe una constante d tal que el espacio necesario para almacenar una configuración de M está acotado por $d \cdot f(n)$

Teorema de Immerman-Szelepcsényi: Demostración

Sea $C_\ell(w)$:

$\{\alpha \mid \alpha$ es una configuración de M que puede ser alcanzada en una cantidad de pasos menor o igual a ℓ en una ejecución de M con entrada w$\}$

Primera observación

Dado $\ell \leq 2^{c \cdot f(n)}$, se puede determinar si $\alpha \in C_\ell(w)$ en espacio $O(f(n))$ en una MT no determinista.

Teorema de Immerman-Szelepcsényi: Demostración

Vale decir, existe una MT no determinista tal que con entrada α:

- funciona en espacio $O(f(n))$
- si en una ejecución se detiene en un estado final, entonces $\alpha \in C_\ell(w)$

¿Cómo se implementa esta MT?

Notación

verificar$(\alpha, C_\ell(w)) =$ *true indica que en la ejecución de la MT se detuvo en un estado final*

- *Para una ejecución que no se detiene en un estado final,* **verificar**$(\alpha, C_\ell(w))$ *no está definido*

Teorema de Immerman-Szelepcsényi: Demostración

Observación fundamental

Se puede calcular $|C_\ell(w)|$ en espacio $O(f(n))$ en una MT no determinista.

Vale decir, existe una MT no determinista tal que con entrada w:

- funciona en espacio $O(f(n))$
- si se detiene en alguna ejecución en un estado final, entonces en la última cinta de trabajo tiene el valor $|C_\ell(w)|$ en binario, antecedido por el símbolo $\vdash$ y precedido de símbolos B

Nótese que $|C_\ell(w)| \leq 2^{c \cdot f(n)}$

- Se necesita de $O(f(n))$ bits para escribir $|C_\ell(w)|$ en binario

Teorema de Immerman-Szelepcsényi: Demostración

Demostración de la observación: Suponga que $m := |C_\ell(w)|$ ya ha sido calculado y está almacenado en una cinta.

Para calcular $|C_{\ell+1}(w)|$ se hace lo siguiente:

total $:= 0$
for each configuración α de M **do**
  adivine m configuraciones $\alpha_1, \ldots, \alpha_m$ de M
  if verificar$(\alpha_i, C_\ell(w)) =$ true para cada $i \in \{1, \ldots, m\}$ **then**
    if existe α_i tal que $\alpha = \alpha_i$ o α es sucesor de α_i en M
      then *total* $:=$ *total* $+ 1$
  else terminar ejecución

¿Cómo se puede implementar esto en espacio $O(f(n))$?

Teorema de Immerman-Szelepcsényi: Demostración

Implementación en espacio $O(f(n))$: Dado orden lexicográfico $<_{\text{lex}}$ sobre las configuraciones de M.

total $:= 0$
for each configuración α de M **do**
  contador $:= |C_\ell(w)|$
  advine una configuración γ de M
  while *contador* > 0 **do**
    $\beta := \gamma$
    if verificar$(\beta, C_\ell(w)) =$ true **then**
      if $\alpha = \beta$ o α es sucesor de β en M **then**
        total $:=$ *total* $+ 1$, *contador* $:= 0$
      else *contador* $:=$ *contador* $- 1$
    else terminar ejecución
    if *contador* > 0 **then**
      adivine una configuración γ de M tal que $\beta <_{\text{lex}} \gamma$

Teorema de Immerman-Szelepcsényi: Demostración

Para terminar: Tenemos que mostrar como determinar si $w \notin L(M)$ en espacio $O(f(n))$.

- Ya tenemos los ingredientes necesarios

Sea α_F la única configuración final de M.

- MT que acepta $\overline{L}$ tiene que mostrar que:

$$\alpha_F \quad \notin \quad C_{2^{c \cdot f(n)}}(w)$$

Un ingrediente importante: $m = |C_{2^{c \cdot f(n)}}(w)|$

- Sabemos como calcular m en espacio $O(f(n))$

Teorema de Immerman-Szelepcsényi: Demostración

Para verificar si $w \notin C_{2^{c \cdot f(n)}}(w)$:

contador $:= m$
advine una configuración γ de M
while *contador* > 0 **do**
  $\beta := \gamma$
  if verificar$(\beta, C_{2^{c \cdot f(n)}}(w)) =$ true **then**
    if $\alpha_F \neq \beta$ **then**
      contador $:=$ *contador* $- 1$
    else terminar ejecución
  else terminar ejecución
  if *contador* > 0 **then**
    adivine una configuración γ de M tal que $\beta <_{\text{lex}} \gamma$
return(true)

Teorema de Immerman-Szelepcsényi: Consecuencias

Un corolario del Teorema de Immerman-Szelepcsényi:

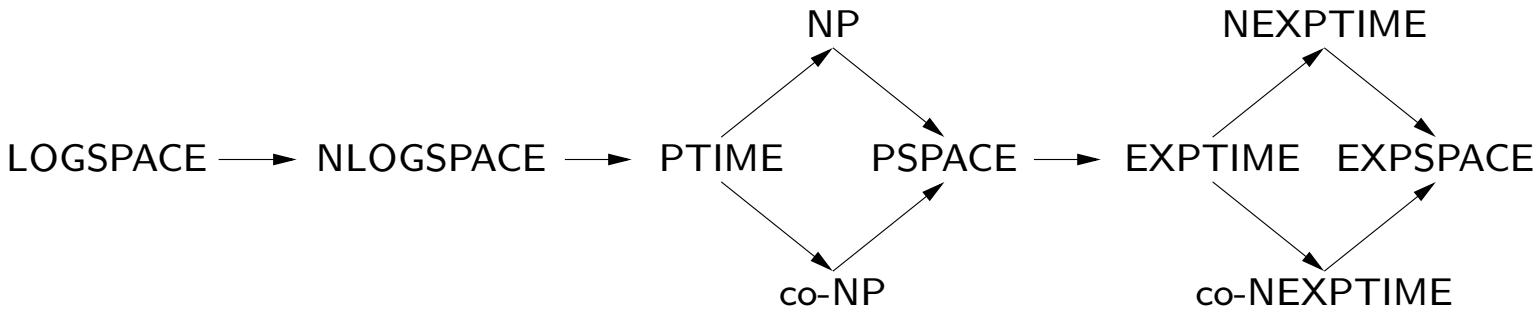
Corolario

$$NLOGSPACE = co\text{-}NLOGSPACE$$

Podemos mejorar aun más la figura ...

Teorema de Immerman-Szelepcsényi: Consecuencias

La nueva versión de la figura:



Todavía nos falta decir más sobre las inclusiones.

- ¿Son algunas de las inclusiones propias?

Paréntesis: Una aplicación en lenguajes formales

Vamos a ver una aplicación del Teorema de Immerman-Szelepcsényi en la área de lenguajes formales.

Un lenguaje libre de contexto es aceptado por una gramática libre de contexto.

- Gramática tiene reglas de la forma $A \to \gamma$.

Ejemplo

$L = \{a^n b^n \mid n \geq 0\}$ es un lenguaje libre de contexto.

Una gramática para L:

$$S \to \varepsilon$$
$$S \to aSb$$

Lenguajes sensibles al contexto

Un lenguaje sensible al contexto es aceptado por una gramática sensible al contexto.

Definición

Gramática sensible al contexto: $G = (V, \Sigma, P, S)$

- V: *conjunto de variables*
- Σ: *alfabeto de entrada* $(V \cap \Sigma = \emptyset)$
- $S \in V$: *variable de inicio*
- P: *conjunto de producciones*
  - *cada producción es de la forma* $S \to \varepsilon$ *ó* $\alpha A \beta \to \alpha \gamma \beta$, *donde* $A \in V$, $\alpha, \beta \in (\Sigma \cup V)^*$ *y* $\gamma \in (\Sigma \cup V)^+$
  - S *no aparece en el lado derecho de ninguna producción*

Lenguajes sensibles al contexto

Dados $u, v \in (\Sigma \cup V)^*$: v puede ser obtenido en G desde u en un paso, denotado como $u \Rightarrow v$, si:

- $u = S$, $v = \varepsilon$ y $S \to \varepsilon$ está en P, o
- $u = u_1 \alpha A \beta u_2$, $v = u_1 \alpha \gamma \beta u_2$ y $\alpha A \beta \to \alpha \gamma \beta$ está en P

$w \in \Sigma^*$ puede ser generado en G si existen w_1, w_2, ..., w_{n-1}, $w_n \in (\Sigma \cup V)^*$, con $n \geq 0$, tales que:

$$S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_{n-1} \Rightarrow w_n \Rightarrow w$$

Definición

$$L(G) \quad = \quad \{w \in \Sigma^* \mid w \textit{ puede ser generado en } G\}$$

Lenguajes sensibles al contexto: Un ejemplo

Un lenguaje L se dice sensible al contexto si existe una gramática sensible al contexto G tal que $L = L(G)$.

Ejemplo

$L = \{a^n b^n c^n \mid n \geq 0\}$ no es un lenguaje libre de contexto, pero si es un lenguaje sensible al contexto.

Una gramática para L:

$$
\begin{array}{rcl}
S & \to & \varepsilon \\
S & \to & S_1 \\
S_1 & \to & aBC \\
S_1 & \to & aS_1BC \\
CB & \to & HB \\
HB & \to & HC
\end{array}
\qquad
\begin{array}{rcl}
HC & \to & BC \\
aB & \to & ab \\
bB & \to & bb \\
bC & \to & bc \\
cC & \to & cc
\end{array}
$$

Para recordar: Clausura bajo intersección y complemento

- Los lenguajes libres de contexto no son cerrados bajo intersección:

$$\begin{aligned} L_1 &= \{a^i b^i c^j \mid i \geq 0 \text{ y } j \geq 0\} \\ L_2 &= \{a^k b^\ell c^\ell \mid k \geq 0 \text{ y } \ell \geq 0\} \end{aligned}$$

  L_1 y L_2 son libres de contexto, pero $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$ no lo es

- Los lenguajes libres de contexto no son cerrados bajo complemento:

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

Propiedades de clausura para los lenguajes sensibles al contexto

¿Son los lenguajes sensibles al contexto cerrados bajo intersección y complemento?

- Para responder esta pregunta usamos complejidad computacional

Teorema

Un lenguaje L es sensible al contexto si y sólo si $L \in NSPACE(n)$.

Antes de demostrar el teorema veamos algunas de sus aplicaciones.

- Lo utilizamos para estudiar propiedades de clausura

Lenguajes sensibles al contexto y complejidad computacional

Corolario

Los lenguajes sensibles al contexto son cerrados bajo intersección y complemento.

Demostración: El corolario es consecuencia del teorema anterior y las siguientes propiedades:

- NSPACE(n) es cerrado bajo intersección
- Si $L \in \text{NSPACE}(n)$, entonces $\overline{L} \in \text{NSPACE}(n)$
  - Por Teorema de Immerman-Szelepcsényi: NSPACE(n) = co-NSPACE(n)

Lenguajes sensibles al contexto y complejidad computacional

El teorema anterior también nos da una idea de que tipo de lenguajes son sensibles al contexto.

Ejercicio

Demuestre que $L = \{a^p \mid p \text{ es un número primo}\}$ es un lenguaje sensible al contexto.

NSPACE(n) = Lenguajes sensibles al contexto

Por demostrar: L es sensible al contexto si y sólo si $L \in \text{NSPACE}(n)$.

($\Rightarrow$) Esta dirección es simple de demostrar.

($\Leftarrow$) Suponga que $L \in \text{NSPACE}(n)$

Existe una MT no determinista $M = (Q, \Sigma, \Gamma, q_0, \delta, F)$ tal que:

- $L = L(M)$
- $s_M(n)$ es $O(n)$

NSPACE(n) $\subseteq$ Lenguajes sensibles al contexto

Tenemos que construir una gramática sensible al contexto $G = (V, \Sigma, P, S)$ tal que $L = L(G)$.

Suponemos que:

- M tiene una cinta de lectura y una de trabajo
- $F = \{q_f\}$ y no existe una tupla en δ de la forma (q_f, a, q', b, X)
- $s_M(n) \leq k \cdot n$ para todo $n \geq 1$, donde $k \in \mathbb{N}$

¿Por qué podemos hacer estas suposiciones?

NSPACE(n) $\subseteq$ Lenguajes sensibles al contexto

¿Cuáles son las variables de G?

- ¿Qué codifican estas variables?

¿Cuáles son las producciones de G?

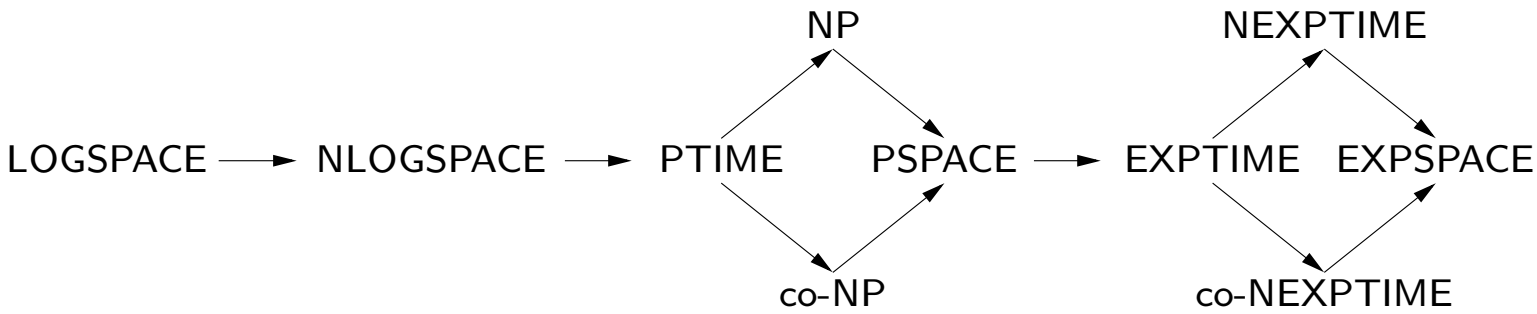
- ¿Cómo se maneja la entrada ε?
- ¿Por qué se necesita de reglas sensibles al contexto?

¿Por qué la demostración no se puede aplicar si $s_M(n)$ es $O(n^2)$?

- ¿Y qué pasa si $s_M(n)$ es $O(n \cdot \log n)$?

Separación de clases de complejidad

Pregunta pendiente: ¿Cuáles de las siguientes inclusiones son propias?



Vamos a ver una técnica general para separar clases de complejidad.

- Está basada en la idea de diagonalización

Separación de clases de complejidad: Supuestos

Primero consideramos clases de complejidad definidas en términos de espacio por MT deterministas.

- Hacemos dos supuestos

Primer supuesto

El alfabeto de entrada es $\Sigma = \{0, 1\}$.

Segundo supuesto

Sólo consideramos MT deterministas con una cinta de trabajo con alfabeto $\Gamma = \{0, 1, \vdash, \mathtt{B}\}$.

¿Por qué podemos hacer estos dos supuestos?

Separación de clases de complejidad: Supuestos

Teorema

Para cada MT determinista M_1 con k cintas de trabajo, existe una MT determinista M_2 con una cinta de trabajo tal que:

- *El alfabeto de la cinta de trabajo de M_2 es $\Gamma = \{0, 1, \vdash, \mathtt{B}\}$*
- $L(M_1) = L(M_2)$
- $s_{M_2}(n)$ *es* $O(s_{M_1}(n))$

Ejercicio

Demuestre el teorema.

Diagonalización: Máquinas de Turing universales

Un ingrediente fundamental para diagonalización: Máquina de Turing universal.

MT universal: Recibe como entrada una MT M y una palabra w.

- Ejecuta M sobre w, y acepta si y sólo si M acepta w

Una MT universal es un computador.

- ¿Cómo recibe M como entrada?
- ¿Cómo ejecuta M sobre w?
- ¿Cuánto espacio necesita para la ejecución?

Máquina de Turing universal: Entrada

Sea $M = (Q, \Sigma, \Gamma, q_0, \delta, F)$ una MT tal que:

- $Q = \{q_0, \ldots, q_m\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{0, 1, \vdash, \mathtt{B}\}$
- $\delta : Q \times \Gamma \times \Gamma \to Q \times \{\leftarrow, \square, \rightarrow\} \times \Gamma \times \{\leftarrow, \square, \rightarrow\}$
- $F = \{q_{i_1}, \ldots, q_{i_k}\}$, con $\{i_1, \ldots, i_k\} \subseteq \{0, \ldots, m\}$

Definimos una codificación $C(M)$ de M como una palabra con alfabeto $\{0, 1\}$.

- Necesitamos esta codificación para dar M como entrada

Máquina de Turing universal: Entrada

Usamos la siguiente representación para los símbolos de M:

símbolo	representación
0	0
1	00
$\vdash$	000
B	0000

símbolo	representación
$\leftarrow$	0
$\square$	00
$\rightarrow$	000

Usando esta notación, definimos $C(M)$ como $w_\delta 1111 w_F$, donde

- $w_F = 0^{i_1+1}10^{i_2+1}1\cdots 10^{i_k+1}$

Máquina de Turing universal: Entrada

- w_δ es de la forma $w_1 11 w_2 11 \cdots 11 w_\ell$, donde cada w_i representa una transición.

  La transición $\delta(q_i, a, b) = (q_j, X, c, Y)$ es representada como:

  $$0^{i+1}10^{k_1}10^{k_2}10^{j+1}10^{k_3}10^{k_4}10^{k_5},$$

  suponiendo que las representaciones de a, b, X, c, Y son 0^{k_1}, 0^{k_2}, 0^{k_3}, 0^{k_4} y 0^{k_5}, respectivamente.

  Por ejemplo, la transición $\delta(q_0, 1, \mathtt{B}) = (q_7, \leftarrow, 0, \rightarrow)$ es representada por la palabra:

  $$0100100001000000010101000$$

Máquina de Turing universal: Entrada

Notación

Máquina de Turing universal: M_u

Entrada de M_u: Una MT M y una palabra w.

- Esto es representado por la palabra $0^{|C(M)|}1C(M)w$

Hay entradas que no representan pares (M, w).

- M_u no acepta estas entradas

Máquina de Turing universal: Propiedad fundamental

Teorema

Existe una MT M_u tal que:

- $L(M_u) = \{x \in \{0,1\}^* \mid x = 0^{|C(M)|}1C(M)w$ *y* M *acepta* $w\}$
- *Existe una constante c_u tal que para cada MT M que satisface que $s_M(n)$ es $\Omega(\log n)$, existe $n_M \geq 0$ tal que para todo $w \in \{0,1\}^*$ con $|w| \geq n_M$:*

$$espacio_{M_u}(0^{|C(M)|}1C(M)w) \quad \leq \quad c_u \cdot espacio_M(w)$$

¿Cómo se construye la máquina M_u?

Máquina de Turing universal: Propiedad fundamental

Ejercicio

Demuestre el teorema utilizando una MT M_u con 4 cintas de trabajo con alfabeto $\Gamma = \{0, 1, \vdash, \mathtt{B}\}$.

- Primera cinta almacena $C(M)$
- Segunda cinta es la cinta de trabajo de M
- Tercera cinta almacena el estado en la ejecución de M
- Cuarta cinta tiene un contador binario que indica la posición de w en la que está parada la cabeza lectora de M
  - Se utiliza para no salir de w

Separación de clases de complejidad: Jerarquía de espacio

Vamos a usar la técnica de diagonalización para demostrar un primer resultado sobre separación de clases de complejidad.

- M_u juega un papel fundamental en esta demostración.

Teorema (Jerarquía de espacio)

Si $f(n)$ *es* $\Omega(\log n)$ *y* $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$, *entonces:*

$$DSPACE(f(n)) \subsetneq DSPACE(g(n))$$

Jerarquía de espacio: Demostración

Demostración: Vamos a definir un lenguaje L tal que $L \notin \text{DSPACE}(f(n))$ y $L \in \text{DSPACE}(g(n))$.

Sea $M^\star$ una MT con 5 cintas de trabajo y alfabeto $\{0, 1, \vdash, \texttt{B}, \#\}$ en estas cintas.

- Definimos L como $L(M^\star)$

Sea w una palabra de largo n. $M^\star$ con entrada w ejecuta los siguientes pasos:

1. Verifica si $w = 1^\ell C(M)$. Si esto es verdad continua funcionando, y en caso contrario para y no acepta.
   - Recuerde que M satisface los supuestos iniciales

Jerarquía de espacio: Demostración

2. Desde la posición 1, $M^\star$ coloca $g(n)$ símbolos $\#$ en las cuatro primeras cintas de trabajo.

3. Desde la posición 1 de la quinta cinta de trabajo, $M^\star$ coloca un símbolo 1 y $g(n)$ símbolos 0.
   - Esto corresponde a $2^{g(n)}$ en binario

4. $M^\star$ simula a M con entrada $1^\ell C(M)$.
   - Utiliza a M_u para hacer esta simulación
   - La cinta de entrada y las 4 primeras cintas de trabajo de $M^\star$ corresponden con las cintas de M_u
   - Sólo hay que hacer una modificación al contador de M_u que está en la cuarta cinta de trabajo

   Por cada paso de M_u, la máquina $M^\star$ decrementa en 1 el contador de la quinta cinta.

Jerarquía de espacio: Demostración

Condición de aceptación de $M^\star$:

(a) Si M_u se sale hacia la derecha del bloque de símbolos $\#$ en alguna de las cuatro primeras cintas de trabajo de $M^\star$, entonces $M^\star$ para y no acepta.

(b) Si no se cumple (a) y el contador en la quinta cinta de trabajo de $M^\star$ llega a 0, entonces $M^\star$ para y acepta.

(c) Si no se cumplen (a) y (b), entonces $M^\star$ acepta w si y sólo si M_u no acepta $0^{|C(M)|}1C(M)1^\ell C(M)$.

- $M^\star$ acepta w si y sólo si M no acepta $1^\ell C(M)$

Jerarquía de espacio: Demostración

Sea $L = L(M^\star)$.

- Una consecuencia de la definición de $M^\star$: $L \in \text{DSPACE}(g(n))$

Lema

$L \notin DSPACE(f(n))$

Por contradicción, suponga que $L \in \text{DSPACE}(f(n))$.

- Existe una MT determinista M_0 tal que $L = L(M_0)$, $s_{M_0}(n)$ es $O(f(n))$ y M_0 satisface los dos supuestos iniciales

Jerarquía de espacio: Demostración

Sabemos que existe n_0 y una constante d tal que $s_{M_0}(n) \leq d \cdot f(n)$ para todo $n \geq n_0$.

Cantidad de configuraciones de M_0 está acotado por:

$$|Q| \cdot (n+2) \cdot d \cdot f(n) \cdot 4^{d \cdot f(n)},$$

suponiendo que Q es el conjunto de estados de M_0.

Como $f(n)$ es $\Omega(\log n)$ y $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$, existe $n_1 \geq n_0$ tal que para todo $n \geq n_1$:

$$\begin{aligned} c_u \cdot d \cdot f(n) &< g(n) \\ |Q| \cdot (n+2) \cdot d \cdot f(n) \cdot 4^{d \cdot f(n)} &< 2^{g(n)} \end{aligned}$$

Jerarquía de espacio: Demostración

Sea $w_0 = 1^\ell C(M_0)$ tal que:

- $|w_0| \geq \text{máx}\{n_1, n_{M_0}\}$
  - Recuerde que n_{M_0} es una constante que depende de M_0, y que aparece en la simulación de M_0 por M_u
- $|C(M_0)| < g(|w_0|)$
- $\log |w_0| < g(|w_0|)$

De esto concluimos que $M^\star$ con entrada $1^\ell C(M_0)$ no satisface la condición (a).

- ¿Por qué?

Tenemos que considerar las condiciones (b) y (c).

Jerarquía de espacio: Demostración

Si se satisface (c), como $L(M^\star) = L = L(M_0)$, obtenemos la siguiente contradicción:

$$M^\star \text{ acepta } 1^\ell C(M_0)$$
si y sólo si
$$M_0 \text{ no acepta } 1^\ell C(M_0)$$
si y sólo si
$$M^\star \text{ no acepta } 1^\ell C(M_0)$$

Jerarquía de espacio: Demostración

Si se satisface (b), entonces el contador de $M^\star$ llegó a 0.

- M_u ejecutó $2^{g(|w_0|)}$ pasos con entrada w_0
- Por definición de $M^\star$: $M^\star$ acepta $1^\ell C(M_0)$

Como $|w_0| \geq n_1$, tenemos que:

$$|Q| \cdot (|w_0| + 2) \cdot d \cdot f(|w_0|) \cdot 4^{d \cdot f(|w_0|)} \quad < \quad 2^{g(|w_0|)}$$

Por lo tanto: En la ejecución de M_u, al menos una configuración de $C(M_0)$ con entrada $1^\ell C(M_0)$ apareció dos veces.

- ¿Por qué?

Jerarquía de espacio: Demostración

Como M_0 es una MT determinista, concluimos que M_0 no para con entrada $1^\ell C(M_0)$.

- M_0 no acepta $1^\ell C(M_0)$.

Como $L(M^\star) = L = L(M_0)$, concluimos que $M^\star$ no acepta $1^\ell C(M_0)$.

- Tenemos una contradicción

$\square$

Jerarquía de espacio: Consecuencias

Usando el teorema de jerarquía de espacio podemos separar clases de complejidad.

Corolario

Para cada número natural $k \geq 1$:

$$DSPACE(n^k) \subsetneq DSPACE(n^{k+1})$$
$$DSPACE(2^{n^k}) \subsetneq DSPACE(2^{n^{k+1}})$$

Ejercicio

Demuestre el corolario.

Jerarquía de espacio: Consecuencias

Corolario

$$LOGSPACE \subsetneq PSPACE \subsetneq EXPSPACE$$

Demostración:

- $\text{LOGSPACE} \subseteq \text{DSPACE}(n) \subsetneq \text{DSPACE}(n^2) \subseteq \text{PSPACE}$
- $\text{PSPACE} \subseteq \text{DSPACE}(2^n) \subsetneq \text{DSPACE}(2^{n^2}) \subseteq \text{EXPSPACE}$

Jerarquía de espacio: Consecuencias

El teorema también puede ser utilizado para separar clases de complejidad definidas en términos de MTs no deterministas.

Corolario

$$NLOGSPACE \subsetneq PSPACE$$

Demostración:

$$\text{NLOGSPACE} \subseteq \text{NSPACE}(n) \subseteq \text{DSPACE}(n^2) \subsetneq \text{DSPACE}(n^3) \subseteq \text{PSPACE}$$

Separación de clases de complejidad: Jerarquía de tiempo

Diagonalización también puede ser usada para separar clases definidas en términos de tiempo por MTs deterministas.

Teorema (Jerarquía de tiempo)

Si $f(n)$ *es* $\Omega(n)$ *y* $\lim_{n\to\infty} \frac{f(n) \cdot \log f(n)}{g(n)} = 0$, *entonces:*

$$DTIME(f(n)) \subsetneq DTIME(g(n))$$

¿Cómo se demuestra este teorema?

- ¿Por qué aparece el término $f(n) \cdot \log f(n)$?

Jerarquía de Tiempo: Consecuencias

Usando el teorema de jerarquía de tiempo podemos separar clases de complejidad.

Corolario

Para cada número natural $k \geq 1$:

$$DTIME(n^k) \subsetneq DTIME(n^{k+1})$$
$$DTIME(2^{n^k}) \subsetneq DTIME(2^{n^{k+1}})$$

Ejercicio

Demuestre el corolario.

Jerarquía de tiempo: Consecuencias

Corolario

$$PTIME \subsetneq EXPTIME$$

Demostración:

- $\text{PTIME} \subseteq \text{DTIME}(2^n) \subsetneq \text{DTIME}(2^{n^2}) \subseteq \text{EXPTIME}$