Teoremas y proposiciones evaluadas en el primer parcial (pero con mis formas de ver la demo)

Proposición 1. *Sea $f : \{0,1\}^* \to \{0,1\}^*$, sea T una función construible en tiempo y sea Γ un alfabeto. Si f es computable en tiempo $T(n)$ por una máquina $M = (\Gamma, Q, \delta)$, entonces f es computable en tiempo $O(\log|\Gamma| \cdot T(n))$ por una máquina $M' = (\Sigma, Q', \delta')$ donde $\Sigma = \{0, 1, \rhd, \square\}$ es el alfabeto estándar.*

Demo:

$|\Gamma|$ es la cantidad de estados que hay originalmente, para codificar cada estado de Q en Q' se usa el alfabeto Σ (binario), por lo cual conlleva $\log_2|\Gamma|$ bits por estado ($\log_2|\Gamma| = \log|\Gamma|$ bits por convención, aunque en big O no importa).

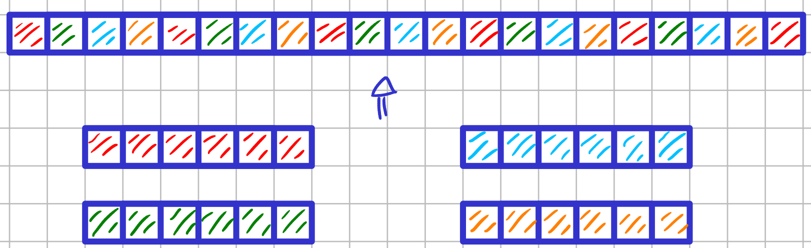
Esto se traslada a δ' también, por lo que lo que antes llevaba un símbolo de Γ ahora lleva $\log|\Gamma|$ de Σ.

Proposición 2. *Sea $f : \{0,1\}^* \to \{0,1\}^*$ y sea T una función construible en tiempo. Si f es computable en tiempo $T(n)$ por una máquina estándar de $k \geq 3$ cintas (entrada, salida y $k-2$ cintas de trabajo), entonces f es computable en tiempo $O(T(n)^2)$ por una máquina de cinta única.*

Demo:

Puedo alternar los símbolos de las k cintas en la cinta única y usar símbolos característicos para indicar dónde está la cabeza de cada cinta.

En la posición i está el caracter $\lceil i/3 \rceil$ de la cinta $i \bmod k$.



Queda en $O(T(n)^2)$ debido a que por cada paso de δ debo primero ubicar cada cabeza para ver qué accionar y después modificar cada cabeza como lo indique δ.

Proposición 3. *Sea $f : \{0,1\}^* \to \{0,1\}^*$ y sea T una función construible en tiempo. Si f es computable en tiempo $T(n)$ por una máquina estándar entonces hay una máquina oblivious que computa f en tiempo $O(T(n)^2)$.*

Demo:

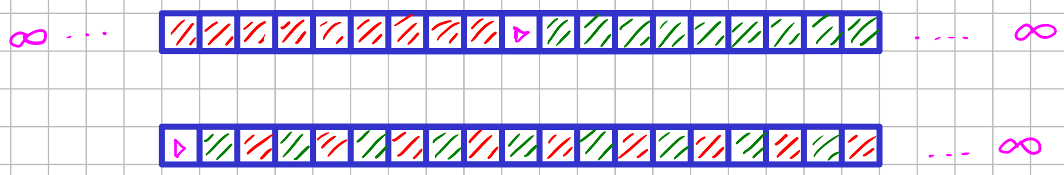
Esto es similar a Prop. 2.

La máquina oblivious se mueve un patrón fijo en cada paso (barre toda la cinta) y cambia la cinta con los cambios que se piden.

Proposición 4. *Sea $f : \{0,1\}^* \to \{0,1\}^*$ y sea T una función construible en tiempo. Si f es computable por una máquina con cintas bi-infinitas en tiempo $T(n)$, entonces f es computable por una máquina estándar en tiempo $O(T(n))$.*

Demo:

Se puede doblar la cinta bi-infinita de manera que rebota en el símbolo $\rhd$ para ver ambos infinitos.



Teorema 1. *[Turing 1936] halt no es computable.*

$$\text{halt}(x) : \begin{cases} 1 & \text{si la } x\text{-ésima máquina con entrada } x \text{ termina } (M_x(x)) \\ 0 & \text{si no} \end{cases}$$

Demo:

Sale por diagonalización, Tengo que:

$$\begin{array}{c|ccc} & M_1 & M_2 & \cdots \\ 1 & M_1(1) & & \\ 2 & & M_2(2) & \\ 3 & & & \ddots \\ \vdots & & & \end{array}$$

Defino entonces M tal q' $M(x)$ termina sii $\text{halt}(x)=0$.

$M(\langle M\rangle)$ termina sii $\text{halt}(\langle M\rangle)=0$ sii $M(\langle M\rangle)$ no termina abs!

Teorema 2. *Existe una máquina U que computa la función $u(\langle i,x\rangle) = M_i(x)$. Más aún, si M_i con entrada x termina en t pasos, entonces U con entrada $\langle i,x\rangle$ termina en $c \cdot t \cdot \log t$ pasos, donde c depende solo de i.*

Demo:

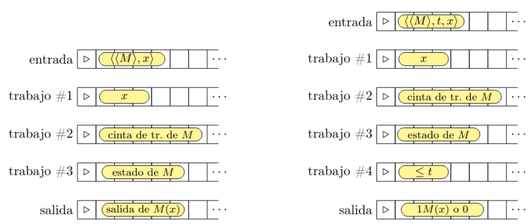


Figura 12: Izquierda: la simulación que hace U (con 3 cintas de trabajo) de M (con una única cinta de trabajo) y entrada x. Derecha: la simulación que hace $\tilde{U}$ (con 4 cintas de trabajo) de M (con una única cinta de trabajo), entrada x hasta el tiempo t.

Pone en cada cinta de trabajo la simulación de la máq. estándar de M. O sea, en #1: La entrada x, #2: Cinta de trabajo de M, #3: Estado de M.
Si #3 $= [q_f]$ termina la ejecución M.
Por cada paso de M busco δ en la entrada de la máquina U.

Teorema 3. *Existe una máquina $\tilde{U}$ que computa la función $\tilde{u}(\langle i,t,x\rangle)$ en tiempo $c \cdot t \cdot \log t$, donde c depende solo de i.*

Demo:

Muy similar a Teo 2 pero con una cinta de trabajo más para llevar registro de i (cant. de pasos en la simulación).

Teorema 4. $\mathbf{P} \subseteq \mathbf{NP}$.

Demo:

Sea $\mathcal{L} \in P$. $\mathcal{L}(M)$, M máq. det. poly.

cadena vacía

Tomás $p(|x|) = 0$. Defino M' det. poly y el certificado $c := \epsilon$, entonces tengo que M' con entrada $\langle x, c \rangle$ copia el comportamiento de M con entrada x.

$$x \in \mathcal{L} \text{ sii } M(x) = 1$$
$$\text{sii } M'(\langle x, c \rangle) = 1$$
$$\text{sii } \exists c.\ c \in \{0,1\}^{\overset{p(|x|)}{0}} \text{ tal q' } M'(\langle x, c \rangle) = 1$$

↳ Def. de NP.

Teorema 6. *Existe una máquina no-determinística NU que tal que NU acepta $(\langle i, x \rangle)$ sii N_i acepta x y si N_i corre en tiempo $T(n)$ entonces $NU(\langle i, x \rangle)$ decide si N_i acepta o rechaza x en $c \cdot T(|x|)$ pasos, donde c depende solo de i.*

Demo: Similar Teo. 2. No tiene el logaritmo porque ...

Teorema 7. *La relación $\leq_p$ es transitiva.*

Demo:

Sea $\mathcal{L} \leq_p \mathcal{L}'$ vía f y $\mathcal{L}' \leq_p \mathcal{L}''$ vía g qvq $\mathcal{L} \leq_p \mathcal{L}''$

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

Ahora qvq $g \circ f$ es computable en tiempo poly a $|x|$.

Sea M_f, M_g tal q' M_f computa f en poly y M_g que computa g en poly.

$M_{g \circ f} : \langle x \rangle$

$y := M_f(x)$ → $O(n^c)$ → A lo sumo su salida es polinomial respecto $n = |x|$

ret $M_g(y)$ → $O((n^c)^d) = O(n^{cd})$ → A lo sumo es poly respecto $|y|$.

Teorema 8. *Si* $\mathbf{NP\text{-}hard} \cap \mathbf{P} \neq \emptyset$, *entonces* $\mathbf{P} = \mathbf{NP}$.

Demo:

Si $\mathcal{L} \in$ NP-hard $\cap P$, entonces puedo tomar cualquier $\mathcal{L}' \in NP$ y reducirlo a $\mathcal{L}$, por lo cual tenga una f computable poly. tal que:

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

Entonces a la $\mathcal{L}'(M_{\mathcal{L}'})$ la decidiré como:

$M_{\mathcal{L}'}: \langle x \rangle$

$y := M_f(x)$ $\rightarrow O(n^c)$

$M_{\mathcal{L}}(y)$ $\rightarrow O(n^{cd})$

Es una máq. det. poly por lo q' $\mathcal{L}' \in P$.

O sea $P = NP$.

Teorema 9. *Si* $\mathcal{L} \in$ **NP-completo**, *entonces* $\mathcal{L} \in \mathbf{P}$ *sii* $\mathbf{P} = \mathbf{NP}$.