Teoremas, Proposiciones y Corolarios

Bander

Julio 2025

Este es un resumen de los teoremas en mis palabras y con sobreexplicación seguramente para así yo lo entiendo. No pretendo reemplazar la documentación oficial, si no que es para tener una explicación para mi yo del futuro.

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 $\Sigma = \{0, 1, \triangleright, \square\}$ es el alfabeto estándar.*

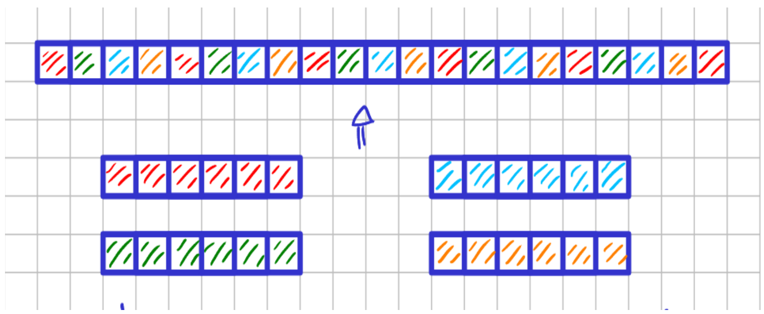
Demostración:

$|\Gamma|$ 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 (por convencion usamos que $log_2|\Gamma| = log|\Gamma|$ bits).[1]
Eso se traslada a δ' también, por lo cual, lo que antes llevaba un símbolo perteneciente a Γ ahora lleva $log_2|\Gamma|$ símbolos de Σ (pej.: $A = 1010$).

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.*

Demostración:

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 $\lceil \frac{i}{k} \rceil$ de la cinta i mod k.



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

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 $O(T(n)^2)$.*

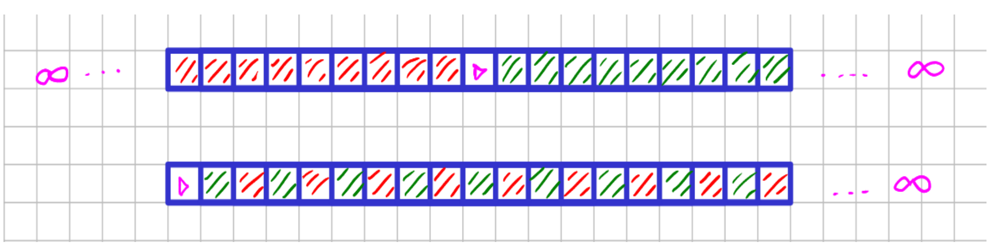
Demostración:

La máquina obliviousse mueve un patrón fijo en cada paso (barre toda la cinta) y cambia la cinta según los cambios que se piden. Es muy similar a la proposición 2.

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

Demostración:

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



[1] La base no importa en la notación big O para los logarítmos ya que difieren entre si por una constante multiplicativa debido a cómo se puede cambiar la base de un logarítmo ($log_b n = \frac{log_k n}{log_k b}$)

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

Recordatorio:

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

Demostración:

Sale por diagonalización. Tengo que:

	M_1	M_2	M_3	$\cdots$
1	$M_1(1)$			
2		$M_2(2)$		
3			$M_3(3)$	
$\vdots$				$\ddots$

Defino entonces M tal que $M(x)$ termina sii $halt(x) = 0$.
$M(\langle M \rangle)$ termina $\iff$ $halt(\langle M \rangle) = 0$ $\iff$ $M(\langle M \rangle)$ no termina. Absurdo!

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.*

Demostración:

Pone en cada cinta de trabajo la simulación de la máquina estándar de M. O sea, en

- #1: La entrada x.
- #2: Cinta de trbajo 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.

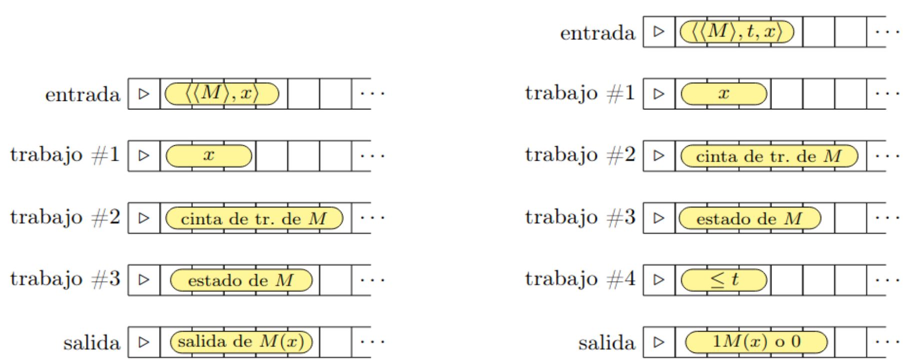


Figura 12: Izquierda: la simulación que hace U (con 3 cintas de trabajo) de M (con una única cinta de trabajo) y entrada x. Derecha: la simulación que hace $\tilde{U}$ (con 4 cintas de trabajo) de M (con una única cinta de trabajo), entrada x hasta el tiempo t.

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

Demostración:

Es muy similar al teorema 2 pero con una cinta más de trabajo para llevar registro de i (cantidad de pasos en la simulación).

Teorema 4: $\mathsf{P} \subseteq \mathsf{NP}$

Demostración:

Sea $\mathcal{L} \in \mathsf{P}$. $\mathcal{L}(M)$, con M una máq. det. que corre en tiempo polinomial.
Tomás el polinomio p de la definición como $p(|x|) = 0$. Defino M' det. y que corre en tiempo poly. y el certificado $c = \varepsilon$ (donde ε es la cadena vacía), entonces tengo que M' con entrada $\langle x, c\rangle$ copia el comportamiento de M con entrada x.

$$\begin{aligned} x \in \mathcal{L} &\iff M(x) = 1 \\ &\iff M'(\langle x, c\rangle) = 1 \\ &\iff \exists c. c \in \{0,1\}^0 \text{ tal que } M'(\langle x, c\rangle) \text{ (Definición de } \mathsf{NP}) \end{aligned}$$

Teorema 5: $\mathsf{NP} = \bigcup_{c\in\mathbb{N}} \mathsf{NDTime}(n^c)$

Demostración:

$\bigcup_{c\in\mathbb{N}} \mathsf{NDTime}(n^c) \subseteq \mathsf{NP}$:
Sea $\mathcal{L} \in \bigcup_{c\in\mathbb{N}} \mathsf{NDTime}(n^c)$ quiero ver que $\mathcal{L} \in \mathsf{NP}$. Sea N una máq. no-det. y p un polinomio tal que N corre en tiempo $p(n)$ y $\mathcal{L}(N)$.
Existe una máq. det. M tal que M con entrada $\langle x, u\rangle$ verifica que u sea la codifcación del cómputo aceptador de N a partir de x. M simula N con entrada x paso a paso, y en cada iteración i usa la primera o la segunda componente de δ según el valor de $u(i)$ para saber qué hacer.
Luego, $x \in \mathcal{L} \iff \exists u \,.\, u \in \{0,1\}^{p(|x|)}$ tal que $M(\langle x, u\rangle) = 1$. Como M corre en tiempo polinomial (ya que es solo recorrer las decisiones en N con u), concluímos que $\mathcal{L} \in \mathsf{NP}$.

$\mathsf{NP} \subseteq \bigcup_{c\in\mathbb{N}} \mathsf{NDTime}(n^c)$:
Sea $\mathcal{L} \in \mathsf{NP}$ quiero ver que $\mathcal{L} \in \bigcup_{c\in\mathbb{N}} \mathsf{NDTime}(n^c)$. Para esto uso la máq. M det. que corre en tiempo poly de la definición y la simulo con una no-det. que hace lo mismo pero inventando el certificado u por lo que te queda que corre en tiempo poly al ser que M corría en tiempo poly.

Teorema 6: *Existe una máquina no-determinística NU 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.*

Demostración:

Similar al teorema 2. No tiene el logarítmo porque puede adivinar no deterministamente la secuencia de elecciones que seguiría N_i para aceptar x.

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

Demostración:

Sea $\mathcal{L} \leq_p \mathcal{L}'$ vía g quiero ver que $\mathcal{L} \leq_p \mathcal{L}''$.

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

Ahora quiero ver que $g \circ f$ es computable en tiempo polinomial respecto $|x|$.
Sea M_f, M_g tal que M_f computa f en tiempo poly y M_g computa g en tiempo poly.

$M_{g\circ f} : \langle x\rangle$
  $y := M_f(x)$ $O(n^c)$ A lo sumo su salida es polinomial respecto n ($n = |x|$)
  return $M_g(y)$ $O((n^c)^d) = O(n^{cd})$ A lo sumo es poly respecto a $|y|$

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

Demostración:

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

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

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

$M_{\mathcal{L}'} : \langle x\rangle$
  $y := M_f(x)$ $O(n^c)$
  $M_{\mathcal{L}}(y)$ $O(n^{cd})$

$M_{\mathcal{L}'}$ es entonces una máq. det. que corre en tiempo poly por lo que $\mathcal{L}' \in \mathsf{P}$. O sea $\mathsf{P} = \mathsf{NP}$, al ser que $\mathcal{L}'$ es genérico.

Teorema 9: *Si* $\mathcal{L} \in \mathsf{NP}$-completo*, entonces* $\mathcal{L} \in \mathsf{P} \iff \mathsf{P} = \mathsf{NP}$.

Demostración:

$\mathcal{L} \in \mathsf{P} \Rightarrow \mathsf{P} = \mathsf{NP}$:
Si $\mathcal{L}$ es NP-completo y a su vez $\mathcal{L} \in \mathsf{P}$, lo único que agrega que $\mathcal{L} \in \mathsf{NP}$-completo es que $\mathcal{L} \in \mathsf{NP}$-hard.[2]
Luego, por el teorema 8 obtengo que $\mathsf{P} = \mathsf{NP}$.

$\mathsf{P} = \mathsf{NP} \Rightarrow \mathcal{L} \in \mathsf{P}$:
Si $\mathsf{P} = \mathsf{NP}$, entonces todos los problemas de NP se pueden resolver en tiempo poly, en particular también todos los NP-completos.

Proposición 6: $\mathsf{TMSAT} \in \mathsf{NP}$-completo.

[2] $\mathcal{L}$ ya era NP, porque $\mathsf{P} \subseteq \mathsf{NP}$