

INF155: Informática Teórica

Clase 21: Códigos de Máquinas de Turing

Aldo Berrios Valenzuela

Jueves 26 de Mayo de 2016

1. Codigos de Máquinas de Turing

Supongamos que la TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$. Podemos suponer:

1. Hay un único estado final.
2. $\Sigma = \{0, 1\}$

Para los puntos (2) y (3): Si “realmente” necesitamos, por ejemplo, Γ un alfabeto mayor, podemos codificar los símbolos en varios bits, y antes de comenzar el proceso “traducir” lo que viene en la cinta.

3. $\Gamma = \{0, 1, B\}$
4. Estados son $\{q_1, q_2, q_3, \dots, q_r\}$; q_1 es inicial, q_2 es final.
5. Simbolos en Γ se numeran:

- $0 \rightarrow 1$
- $1 \rightarrow 2$
- $B \rightarrow 3$

6. Direcciones:

- Izquierda $\rightarrow 1$
- Derecha $\rightarrow 2$

7. Podemos representar la función δ :

$$\delta(q, a) = \{(p_1, a_1, d_1), (p_2, a_2, d_2), \dots\}$$

Con $q \in Q$, $a \in \Gamma$, $p_i \in Q$, $a_i \in \Gamma$ y $d_i \in \{\text{izq}, \text{der}\}$

8. En vez de escribir el conjunto damos cada posibilidad, o sea listar los quintetos:

$$q, a, p_i, a_i, d_i$$

Estos podemos representarlos:

$$0^q 10^a 10^{p_i} 10^{a_i} 10^{d_i} \tag{1.1}$$

Para representar M , listamos los quintetos de todos sus movidas posibles, separados por 11.

Una secuencia de $\{0, 1\}$ que no tenga el “formato correcto” (no comienza en 0, no están en $\mathcal{L}((0^+ 10^+ 10^+ 10^+ 10^+ 11)^* 0^+ 10^+ 10^+ 10^+ 10^+)$) consideramos que representa la TM de la Figura 1.



Figura 1: Máquina de Turing que no cumple con el formato $\mathcal{L}((0^+10^+10^+10^+10^+11)^*0^+10^+10^+10^+10^+)$

\rightsquigarrow Tenemos una función de $\{0,1\}^* \rightarrow \text{TM}$.

Podemos numerar los elementos de $\{0,1\}^*$:

$\epsilon \rightarrow 1$
 $0 \rightarrow 2$
 $1 \rightarrow 3$
 $00 \rightarrow 4$
 $01 \rightarrow 5$
 $10 \rightarrow 6$
 $11 \rightarrow 7$
 \vdots

Esto es una biyección entre \mathbb{N} y Σ^* .

Interesa representar: $\langle M, \sigma \rangle$, donde M es una TM, $\sigma \in \{0,1\}^*$ son datos de entrada. Una opción es:

$$\langle M \rangle 111\sigma \quad (1.2)$$

En otras palabras, (1.2) nos dice que separamos la máquina de Turing M de los datos de entrada σ a través de una secuencia de tres 1's.

Por otro lado, la relación (1.2) me permite hablar de la TM número i , M_i , y los datos de entrada número j , σ_j .

1.1. Punto Central

Definición 1.1. Definimos el lenguaje diagonal L_d , como:

$$L_d = \{i : \sigma_i \notin \mathcal{L}(M_i)\} \quad (1.3)$$

Con dibujitos (Cuadro 1):

		$\sigma_i \rightarrow$				
		1	2	3	4	5
$M_i \downarrow$	1	1	0	1
	2	0	0	1
	3	1	0	0
	4	\vdots	\vdots	\vdots	1	...
	5	\vdots	\vdots	\vdots	\vdots	\ddots

Cuadro 1: Los números rojos representan la diagonal de la matriz.

Observaciones importantes que acotar de la matriz (Cuadro 1):

- Si $a_{ij} = 0$, entonces la Máquina de Turing M_i no acepta σ_j .
- Si $a_{ij} = 1$, entonces la Máquina de Turing M_i acepta σ_j .

¿Hay una máquina de Turing que acepte L_d ?

Teorema 1.1. No hay máquina de turing que acepte L_d .

Demostración. Por contradicción. Supongamos que M acepta L_d . Entonces $M = M_i$ para algún i . Entonces M_i acepta σ_i , pero eso es que $i \notin L_d$. Lo cual es una contradicción. \square

L_d en el fondo son máquinas de turing que no se aceptan a si mismas \rightarrow problema de “Hola mundo”.

La demostración del Teorema 1.1 era algo bastante esperado, ya que hay más lenguajes que máquinas de turing, por lo tanto, hay lenguajes que no pueden ser representados por TM.¹

Definición 1.2. Un lenguaje se llama recursivamente enumerable (R.E.) si hay una TM que lo acepte.

- Recursivo: si hay una TM **que siempre se detiene** que lo acepte.
- Decidible: si es recursivo.
- No decidible: si no es recursivo.

Observación Nótese que L_d al ser *no decidible*, se tiene que tampoco es *recursivamente enumerable*.

¹Para entender de mejor manera esto, ver el capítulo de “Numerabilidad” del apunte de Fundamentos de Informática