Autómatas y Lenguajes Formales
Tarea 8

Alumnos:
Torres Partida Karen Larissa
Altamirano Niño Luis Enrique

22 de junio de 2020

1. Diseñe una máquina de Turing determinista, especificando cada uno de sus siete elementos a detalle, de manera que reconozca el lenguaje $A = \{a^{2^n} | n \geq 0\}$, el lenguaje que consiste de todas las cadenas de a's cuya longitud es una potencia de 2.

   Respuesta:

   Tenemos la siguiente máquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F)$ dónde:

   - $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_f\}$
   - $\Sigma = \{a\}$
   - $\Gamma = \{a, X, Y, \sqcup\}$
   - $F = \{q_f\}$
     Y definimos la función de transición como sigue:

   $$\begin{array}{ll} \delta(q_0, a) = (q_1, a, R) & \delta(q_1, \sqcup) = (q_f, \sqcup, R) \\ \delta(q_1, a) = (q_2, a, L) & \delta(q_2, a) = (q_3, X, R) \\ \delta(q_2, Y) = (q_6, Y, R) & \delta(q_3, a) = (q_3, a, R) \\ \delta(q_3, \sqcup) = (q_4, \sqcup, L) & \delta(q_4, a) = (q_5, Y, L) \\ \delta(q_5, a) = (q_5, a, L) & \delta(q_5, X) = (q_2, X, R) \\ \delta(q_3, Y) = (q_4, Y, L) & \delta(q_4, \sqcup) = (q_6, \sqcup, R) \\ \delta(q_6, X) = (q_6, a, R) & \delta(q_6, Y) = (q_6, \sqcup, R) \\ \delta(q_6, \sqcup) = (q_6, \sqcup, L) & \delta(q_6, a) = (q_7, a, L) \\ \delta(q_7, a) = (q_7, a, L) & \delta(q_7, \sqcup) = (q_0, \sqcup, R) \\ \delta(q_7, x) = (q_7, a, L) & \end{array}$$

2. Para los propósitos de este problema, si está describiendo una Máquina de Turing, no es necesario que especifique los elementos de la tupla. Una descripción correcta, precisa y clara es suficiente. Muestre que los lenguajes recursivamente enumerables son cerrados bajo la operación de reversa.

   Respuesta:

   Sea L el lenguaje recursivamente enumerable que es reconocido por una MT M. Entonces, construimos una MT M' que tiene una cinta que procesa una cadena de entrada w y devuelve w^R ahora tomamos esta cadena y la pasamos por M, dónde la cadena puede ser aceptada o no.
   Entonces L^R es reconocido por una MT y por lo tanto L^R es recursivamente enumerable. Por lo tanto los lenguajes recursivamente enumerables son cerrados bajo la operación reversa.

3. Suponga que T es una máquina de Turing que acepta el lenguaje L. Describa como construir una máquina de Turing no determinista que acepte L^*.

   Respuesta:

   Vamos a definir a la MT no-determinista T^* que reconoce L^*. Teniendo la cadena de entrada $w = w_1 w_2 ... w_n$ copiaremos los primeros k símbolos con $k \leq n$ en una segunda cinta, ejecutando T en esta cinta puede haber tres casos:
   1. Si la cadena es aceptada por T y no hay más símbolos por procesar ($k = n$), entonces T^* acepta la cadena.
   2. Si la cadena es aceptada por T y todavía hay símbolos por procesar ($k < n$), entonces se repiten los pasos con $w' = w_{k+1} w_{k+2} ... w_n$
   3. Si la cadena no es aceptada por T, entonces se rechaza la cadena.

   Dado que T^* es no-determinista, basta con que exista un camino donde sí se acepte la cadena sin importar que se rechace en los demás.

4. Una máquina de Turing con movimiento neutro es una MT determinista convencional donde los movimientos permitidos son izquierda (L), derecha (D) y neutro (N). Realizar un movimiento neutro quiere decir que la cabeza lectora permanece inmóvil al realizar una transición. Así, su función de transición es $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, N, D\}$. Desmuestre que esta variante de la máquina de Turing es equivalente a su definición convencional ¿Cómo es que se puede simular un movimiento neutro a través de movimientos L y R?

   Respuesta:

   Sea $M_N = (Q_N, \Sigma, \Gamma, \delta_{M_N}, q_0, \sqcup, F)$ una MT con movimiento neutro y $M = (Q_M, \Sigma, \Gamma, \delta_M, q_0, \sqcup, F)$ la MT equivalente sin movimientos neutros.
   Si en M_N tenemos $\delta(q, w) = (p, y, N)$ dónde $q, p \in Q_N$, $w, y \in \Gamma$, entonces en M tendriamos $\delta(q, w) = (v, y, L)$, dónde v es un nuevo estado en M, $y, w \in \Gamma$, esto para cada transición con movimiento neutro, y además tendriamos que $d(v, x_i) = (q, x_i, R), \forall x_i \in \Gamma$. De esta manera logramos simular los movimientos neutros en una MT convencional, por lo tanto ambas definiciones de la MT son equivalentes.

5. Demuestre que el lenguaje $\overline{L_d}$ es recursivamente enumerable pero no es recursivo.

   Demostración

   Por el *Teorema 4.2* L_d no es recursivamente enumerable, por lo que se sigue que L_d no es recursivo, lo anterior es equivalente a que $\overline{\overline{L_d}}$ no es recursivo, ahora usando la forma contrapuesta del *Teorema 1.1* tenemos que entonces $\overline{L_d}$ no es recursivo. Además se sabe que $L_u \prec \overline{L_d}$, entonces $\overline{L_d}$ es recursivamente enumerable pues es reconocido por la máquina de Turing universal M_u.
   Por lo tanto el lenguaje $\overline{L_d}$ es recursivamente enumerable pero no es recursivo.

6. Por medio de una reducción demuestre que el lenguaje $L^{\diamondsuit} = \{N | N \text{ es una MT y } L(N) \text{ es un lenguaje recursivo}\}$ no es recursivo.

   Respuesta:

   Para efectuar la reducción necesitamos escoger un lenguaje que no sea recursivo. Tomaremos

   $$\overline{L_H} = \{< M, w > | M \text{ no se detiene al procesar } w \text{ como entrada}\}$$

el cual no es recursivo. Buscamos $\overline{L_H} \prec L^{\Diamond}$. Por corolario 5.5 vamos a concluir que $L^{\Diamond}$ no es recursivo. Definimos la reducción $f(< M, w >) = M'$ como la MT sobre cualquier entrada x que realiza lo siguiente:

a) Guarda x en una pista de su cinta.

b) Escribe w en una pista diferente.

c) Ejecuta M sobre w.

d) Si M se detiene, entonces M' realiza lo necesario para aceptar a x si es de la forma $\{< M, w > | w \in L(M)\}$.

Lo anterior describe el comportamiento de M', por lo que $f(< M, w >)$ lo calcula una MT total.

Obs. 1 Si M no se detiene sobre w, no se ejecutará el inciso *d)* y M' reconocerá a $\emptyset$.
Obs. 2 Si M se detiene sobre w, M' acepta el lenguaje $\{< M, w > | w \in L(M)\}$

Así:

$$L(M') = \begin{cases} \emptyset & \text{si } M \text{ no se detiene sobre } w \\ \{< M, w > | w \in L(M)\} & \text{si } M \text{ se detiene sobre } w \end{cases}$$

$< M, w > \in \overline{L_H} \to L(M') = \emptyset \to M' \in L^{\Diamond}$ esto es verdad pues fácilmente se puede describir una MT total para el vacío, tal que esta rechaza cualquier entrada.

$< M, w > \notin \overline{L_H} \to L(M') \to \{< M, w > | w \in L(M)\} \to M' \notin L^{\Diamond}$

Por lo tanto $< M, w > \in \overline{L_H} \leftrightarrow f(< M, w >) = M' \in L^{\Diamond}$

Por el corolario 5.5 de la nota 14, $L^{\Diamond}$ no es recursivo.