Universidad Nacional Autónoma de México

Facultad de Ciencias

Automatas y Lenguajes Formales

Tarea 03

Arriaga Santana Estela Monserrat
Castañon Maldonado Carlos Emilio
Fernández Blancas Melissa Lizbeth

1 **Define una máquina de Turing determinista de una sola cinta que procese el lenguaje $L = \{0^n1^n : n \geq 0\}$ y prueba que acepta 0011 y que se detiene sin aceptar con 011 y 001.**

Sea una maquina de Turing con las siguientes características:

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{0, 1, X, Y, B\}$
- q_0 = Estado Inicial
- $\{q_4\}$ = Estado Final

	0	**1**	**Y**	**X**	**B**
q_0	$(q_1, 0, R)$	-	-	-	(q_4, B, R)
q_1	$(q_1, 0, R)$	(q_2, X, L)	-	-	-
q_2	(q_2, Y, R)	-	-	(q_3, B, L)	-
q_3	-	-	(q_0, B, L)	-	-
q_4	-	-	-	-	-

A partir de la maquina de Turing descrita, veremos si 0011 es aceptada

$$q_0 0011 \vdash 0q_1 011 \vdash 00q_1 11 \vdash 0q_2 0X1 \vdash 0Yq_2 X1 \vdash 0q_3 Y1 \vdash q_0 01 \vdash 0q_1 1 \vdash q_2 0X \vdash Yq_2 X \vdash q_3 Y \vdash q_0 B \vdash Bq_4$$

Como hemos llegado al estado final podemos decir que **la cadena es aceptada.**

A partir de la maquina de Turing descrita, veremos si 001 es aceptada

$$q_0 001 \vdash 0q_1 01 \vdash 00q_1 1 \vdash 0q_2 0X \vdash 0Yq_2 X \vdash 0q_3 Y \vdash q_0 0 \vdash 0q_1 B$$

Como podemos observar la cadena no es aceptada ya que cuando llegamos al estado (q_1) no existe alguna transición para B, con esto podemos decir que **la cadena no es aceptada.**

A partir de la maquina de Turing descrita, veremos si 011 es aceptada

$$q_0 011 \vdash 0q_1 11 \vdash q_2 0X1 \vdash Yq_2 X1 \vdash q_3 Y1 \vdash q_0 1$$

Como en q_0 no hay transición alguna para 1, podemos decir que **la cadena no es aceptada.**

2 **Dada la siguiente máquina de Turing descrita por la tabla de transición, con $F = \{q_2\}$:**

	0	**1**	**B**
q_0	$(q_1, 0, R)$	$(q_0, 1, R)$	-
q_1	$(q_1, 0, R)$	$(q_0, 1, L)$	(q_2, B, R)
q_2	-	-	-

Para las siguientes cadenas:

- Generar cada una de sus descripciones instantáneas en M
- Determinar si son o no aceptadas por el lenguaje
- Determinar si la máquina de Turing se detiene o no:

(a) 11100

$$q_0 11100 \vdash 1q_0 1100 \vdash 11q_0 100 \vdash 111q_0 00 \vdash 1110q_1 0 \vdash 11100q_1 \vdash 11100Bq_2$$

Como se pudo llegar al estado final, entonces la cadena es aceptada por el lenguaje de la máquina y al no haber transiciones en q_2, la máquina se detiene.

(b) 001110010

$$q_0 001110010 \vdash 0q_1 01110010 \vdash 00q_1 1110010 \vdash 0q_0 01110010 \vdash 00q_1 1110010 \vdash 0q_0 01110010 \vdash ...$$

Esta cadena no es aceptada por el lenguaje, pues la máquina entra en un bucle haciendo que la máquina no se detenga y que no sea posible llegar al estado final.

(c) 100

$$q_0 100 \vdash 1q_0 00 \vdash 10q_1 0 \vdash 100q_1 \vdash 100Bq_2$$

Como se pudo llegar al estado final, entonces la cadena es aceptada por el lenguaje de la máquina y al no haber transiciones en q_2, la máquina se detiene.

(d) 11111

$$q_0 11111 \vdash 1q_0 1111 \vdash 11q_0 111 \vdash 111q_0 11 \vdash 1111q_0 1 \vdash 11111q_0$$

Esta cadena no es aceptada por el lenguaje, pues no pudimos llegar al estado final. Además la máquina de Turing se detiene.

3 **Dada la máquina de Turing descrita por la siguiente tabla de transiciones donde $[q_1, B]$ es estado final:**

	0	**1**	**B**
$[q_0, B]$	$([q_1, 0], 0, R)$	$([q_1, 1], 1, R)$	-
$[q_1, 0]$	-	$([q_1, 0], 1, R)$	$([q_1, B], B, R)$
$[q_1, 1]$	$([q_1, 1], 0, R)$	-	$([q_1, B], B, R)$
$[q_1, B]$	-	-	-

Realizar las descripciones instantáneas para las siguiente cadenas y determina si son aceptadas:

(a) 0111

$$[q_0, B]0111 \vdash 0[q_1, 0]111 \vdash 01[q_1, 0]11 \vdash 011[q_1, 0]1 \vdash 0111[q_1, 0] \vdash 0111B[q_1, B]$$

Esta cadena es aceptada por la máquina debido a que pudimos llegar al estado final.

(b) 100

$$[q_0, B]100 \vdash 1[q_1, 1]00 \vdash 10[q_1, 1]0 \vdash 100[q_1, 1] \vdash 100B[q_1, B]$$

Esta cadena es aceptada por la máquina debido a que pudimos llegar al estado final.

(c) 0

$$[q_0, B]0 \vdash 0[q_1, 0] \vdash 0B[q_1, B]$$

Esta cadena es aceptada por la máquina debido a que pudimos llegar al estado final.

(d) 101

$$[q_0, B]101 \vdash 1[q_1, 1]01 \vdash 10[q_1, 1]1$$

Esta cadena no es aceptada por la máquina de Turing, ya que sólo pudimos llegar a $[q_1, 1]$, que no es estado final.

4 **Mostrar que si L es un lenguaje recursivo, su complemento L^c es también recursivo.**

Sea L un lenguaje recursivo, entonces existe una máquina de Turing M tal que $L = L(M)$ que siempre se detiene. Sea $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, entonces podemos construir una máquina de Turing M' tal que, si $w \in \Sigma^*$ y w es aceptada por M, entonces w no es aceptada por M' y si w no es aceptada por M, entonces w es aceptada por M'.

La construcción de M' la haremos de la siguiente manera. Sea $p \notin Q$ un nuevo estado, entonces haremos $M' = (Q', \Sigma, \Gamma, \delta', q_0, B, F')$ tal que $Q' = Q \cup \{p\}$, $F' = \{p\}$ y δ' se construye modificando las transiciones de la máquina M de la siguiente manera:

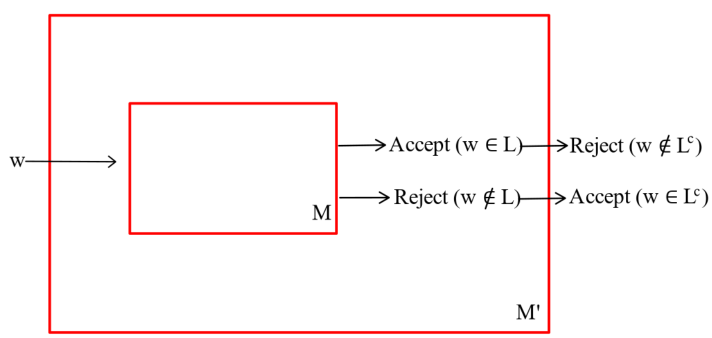
- Los estados de aceptación de M serán los estados en los que M' se detenga sin aceptar.
- Los estados en los que M se detiene sin aceptar, se agregará una transición del estado en que se detiene hacia p.
- M' va a simular las otras transiciones de M.

5 **Demostrar que si L es un lenguaje recursivo, entonces su complemento L^c también es recursivo.**

Tomando la construcción de M' que hicimos en el inciso anterior, tenemos que:

- Si $w \in L$, entonces $w \notin L^c$. Además, w llega a un estado de aceptación en M, por lo que M' se detiene sin aceptar y con esto, $w \in L(M')$.
- Si $w \notin L$, entonces $w \in L^c$. Además, M se detiene sin aceptar a w, por lo que w llega al estado final de M' y w es aceptada por M', por lo que $w \in L(M')$.

Con esto tenemos entonces los siguiente:



Es decir:

- Si $w \in L$, entonces $w \notin L^c$, así que M acepta y M' rechaza.
- Si $w \notin L$, entonces $w \in L^c$, así que M rechaza y M' acepta.

Tomando en cuenta lo anterior, tenemos que $L(M') = L^c$ y como es posible construir M' que siempre se detiene con $w \in \Sigma^*$, entonces L^c es un lenguaje recursivo.

6 **Demostrar que si G_1 y G_2 son dos gramáticas libres de contexto, el problema de determinar $L(G_1) \cap L(G_2) = \emptyset$ es indecidible**
(HINT: Reduce el problema al PCP usando los lenguajes L_A y L_B).

Reducimos este problema al PCP. Para esto, dada una instancia del PCP con $A = [w_1, ..., w_k]$ y $B = [x_1, ..., x_k]$ construimos dos gramáticas G_A y G_B.
Tomamos un conjunto de símbolos $a_1, ..., a_k$ que representan elecciones de pares de cadenas en una instancias del PCP. Así a_i representa la elección de w_i de A y de x_i de B.
La gramática para A es la siguiente:

$$S_1 \to w_1 S_1 a_1 \mid ... \mid w_k S_1 a_k \mid w_1 a_1 \mid ... \mid w_k a_k$$

Construimos una gramática similar para B.

$$S_2 \to x_1 S_2 a_1 \mid ... \mid x_k S_2 a_k \mid x_1 a_1 \mid ... \mid x_k a_k$$

Notemos que la instancia dada del PCP tiene solución si y solo si $L(G_A) \cap L(G_B) \neq \emptyset$.
Por lo tanto, este problema se reduce al PCP y concluimos que es indecidible.

7 **Demostrar que si G_1 y G_2 son dos gramáticas libres de contexto, el problema de determinar $L(G_1) = L(G_2)$ es indecidible.**

Reducimos este problema al PCP. Para esto, dada una instancia del PCP con $A = [w_1, ..., w_k]$ y $B = [x_1, ..., x_k]$ construimos las gramáticas G_A y G_B de manera que para toda $a \in A$ obtenemos una producción de G_A $S_1 \to a$ y para toda $b \in B$ obtenemos una producción de G_B $S_2 \to b$.

Con esto, tenemos las gramáticas G_A y G_B que producen las cadenas de A y B respectivamente.

Entonces la instancia dada del PCP tiene solución si y sólo si $L(G_A) = L(G_B)$.
Por lo tanto, es una reducción al PCP y el problema es indecidible.