

# Práctica 3

## 1. Construir máquinas de Turing que acepten los siguientes lenguajes

a.  $L_1 = \Sigma^*$

Asumo  $\Sigma = \{0, 1\}$

Esta MT acepta cualquier cadena posible del alfabeto.

$q\theta, \theta$   
 $qA, \theta, D$

$q\theta, 1$   
 $qA, 1, D$

$q\theta, B$   
 $qA, B, D$

b.  $L_2 = \{\lambda\}$

Asumo  $\Sigma = \{0, 1\}$

Esta MT acepta únicamente la cadena vacía.

$q\theta, \theta$   
 $qR, \theta, D$

$q\theta, 1$   
 $qR, 1, D$

$q\theta, B$   
 $qA, B, D$

c.  $L_3 = \emptyset$

Asumo  $\Sigma = \{0, 1\}$

Esta MT no acepta ninguna cadena, ni siquiera la cadena vacía.

$q\theta, \theta$

$qR, \theta, D$

$q\theta, 1$

$qR, 1, D$

$q\theta, B$

$qR, B, D$

d.  $L_4 = \{0^n 1^{2n} \mid n \geq 0\}$

Asumo  $\Sigma = \{0, 1\}$

Esta MT acepta la cadena vacía y además cadenas que tienen el doble de unos que de ceros. Sus primeras cadenas serían:  $\lambda, 011, 001111, 00011111, \dots$

Para la solución, la idea es que por cada cero que se borra, se deben borrar dos unos. Si en algún momento no se puede cumplir esta condición, se rechaza la cadena.

```
// Si es blanco se acepta. Si empieza con 1 se rechaza.  
q0, B  
qA, B, D  
  
q0, 1  
qR, 1, D  
  
q0, 0  
q1, B, D  
  
// Busco todos los ceros. Si hay blanco después de los ceros, se rechaza.  
q1, 0  
q1, 0, D  
  
q1, B  
qR, B, D  
  
q1, 1  
q2, 1, D  
  
// Voy al final de la cadena (después de todos los unos). Si hay ceros después de los unos, se rechaza.  
q2, 1  
q2, 1, D  
  
q2, 0  
qR, 0, D  
  
q2, B  
q3, B, I  
  
// Borro un solo uno  
q3, 1  
q4, B, I  
  
// Borro otro uno  
q4, 1  
q5, B, I  
  
q4, 0  
qR, 0, I  
  
// Voy al principio de la cadena y borro un cero  
q5, 1
```

q5, 1, I

q5, 0

q5, 0, I

q5, B

q0, B, D

e.  $L_5 = \{a^n b^n c^n \mid n \geq 0\}$

Asumo  $\Sigma = \{a, b, c\}$

Esta MT acepta la cadena vacía y además cadenas que tienen la misma cantidad de  $as$ ,  $bs$  y  $cs$ . Sus primeras cadenas serían:  $\lambda, abc, aabbcc, aaabbbccc, \dots$

...

f.  $L_6 = \{a^n b^m c^k \mid k = n + m, n, m \geq 1\}$

Asumo  $\Sigma = \{a, b, c\}$

Esta MT acepta cadenas que tienen una cantidad de  $cs$  igual a la suma de la cantidad de  $as$  y  $bs$ , con al menos una  $a$  y una  $b$ . Sus primeras cadenas serían:

$abcc, aabccc, abbccc, aabbcccc, aaabbccccc, aabbccccc, \dots$

...

g.  $L_7 = \{ww^R \mid w \in \{0, 1\}^*\}$ , donde  $w^R$  es el reverso de  $w$

Asumo  $\Sigma = \{0, 1\}$

Esta MT acepta cadenas palíndromes y además su cantidad de símbolos es par. Sus primeras cadenas serían:  $\lambda, 00, 11, 1001, 0110, 101101, \dots$

Para la solución, se debe chequear que cada símbolo del inicio de la cadena sea exactamente igual al del final de la cadena, y borrar ambos. Se repite el proceso hasta que no queden más símbolos (cadena vacía) o hasta que quede un solo símbolo (cadena de longitud impar), en cuyo caso se rechaza la cadena.

```
// Se acepta lambda. Si empieza con 0 hay que buscar un 0 al final, si empieza con 1 hay que bus
q0, B
qA, B, D

q0, 0
q1, B, D

q0, 1
q2, B, D

// Buscar el 0 al final
q1, 0
q1, 0, D

q1, 1
q1, 1, D

q1, B
q3, B, I

// Borrar el 0 del final
q3, 1
qR, 1, D

q3, 0
q5, B, I

q3, B
qR, B, I

// Buscar el 1 al final
q2, 0
q2, 0, D

q2, 1
q2, 1, D

q2, B
q4, B, I

// Borrar el 1 del final
q4, 0
qR, 0, D
```

```
q4, 1  
q5, B, I
```

```
q4, B  
qR, B, I
```

```
// Volver al inicio de la cadena
```

```
q5, 1  
q5, 1, I
```

```
q5, 0  
q5, 0, I
```

```
q5, B  
q0, B, D
```

$$\text{h. } L_8 = L_7 \cup \{w0w^R \mid w \in \{0,1\}^*\} \cup \{w1w^R \mid w \in \{0,1\}^*\}$$

Asumo  $\Sigma = \{0,1\}$

Esta MT acepta cadenas palíndromes que pueden tener un uno o un cero en el medio, por lo tanto puede tener una cantidad impar de símbolos. Sus primeras cadenas serían:

$\lambda, 00, 11, 1001, 0110, 101101, \dots, 010, 000, 101, 111, 10001, 10101, \dots$

La solución es muy similar al anterior, pero al "volver al inicio de la cadena", hay que hacer algunos cambios.

```
// Se acepta lambda. Si empieza con 0 hay que buscar un 0 al final, si empieza con 1 hay que bus
q0, B
qA, B, D

q0, 0
q1, B, D

q0, 1
q2, B, D

// Buscar el 0 al final
q1, 0
q1, 0, D

q1, 1
q1, 1, D

q1, B
q3, B, I

// Borrar el 0 del final
q3, 1
qR, 1, D

q3, 0
q5, B, I

q3, B
qR, B, I

// Buscar el 1 al final
q2, 0
q2, 0, D

q2, 1
q2, 1, D

q2, B
q4, B, I

// Borrar el 1 del final
q4, 0
qR, 0, D
```

```
q4, 1
q5, B, I

q4, B
qR, B, I

// Volver al inicio de la cadena (cambios a partir de acá)
// Nos movemos un solo paso hacia la izquierda
q5, 1
q6, 1, I

q5, 0
q6, 0, I

q5, B
q0, B, D

// Si había un solo cero o uno restante, se acepta
q6, 1
q7, 1, I

q6, 0
q7, 0, I

q6, B
qA, B, D

// Vamos al inicio de la cadena
q7, 0
q7, 0, I

q7, 1
q7, 1, I

q7, B
q0, B, D
```

## 2. Construya una Máquina de Turing de 2 cintas que implemente un contador binario en la segunda cinta para

## **contabilizar la cantidad de letras $a$ que aparecen en el input de la primera cinta. Con $\Sigma = \{a, b\}; \Gamma = \{a, b, 0, 1, B\}$**

Se debe sumar la cantidad de letras  $a$  que aparecen en la primera cinta y escribir el resultado en binario en la segunda cinta. Por ejemplo, *aaabbaba* tiene 5 letras  $a$ , que en binario es 101, por lo que la segunda cinta debe contener esa cadena al finalizar la ejecución.

```
// Ignorar las 'b's en la primera cinta  
q0, b, B  
q0, b, D, B, S
```

```
q0, b, 0  
q0, b, D, 0, S
```

```
q0, b, 1  
q0, b, D, 1, S
```

```
// Fin de la cadena  
q0, B, B  
qd, B, S, B, S
```

```
q0, B, 0  
qd, B, S, 0, S
```

```
q0, B, 1  
qd, B, S, 1, S
```

```
// Encontrar una 'a' en la primera cinta  
q0, a, B  
q1, a, S, B, S
```

```
q0, a, 0  
q1, a, S, 0, S
```

```
q0, a, 1  
q1, a, S, 1, S
```

```
// Suma binaria sin overflow  
q1, a, B  
q0, a, D, 1, S
```

```
q1, a, 0  
q0, a, D, 1, S
```

```
q1, a, 1  
q2, a, S, 0, I
```

```
// Suma binaria con overflow  
q2, a, 1  
q2, a, S, 0, I
```

```

q2, a, B
q3, a, S, 1, D

q2, a, 0
q3, a, S, 1, D

// Volvemos al final de la segunda cinta
q3, a, 0
q3, a, S, 0, D

q3, a, B
q0, a, D, B, I

```

**3. Sea  $M$  una máquina de Turing del modelo DIS. ¿Existe una máquina de Turing  $M'$  equivalente a  $M$  que comience con el cabezal apuntando a cualquier celda de la cinta? Note que  $M'$  puede apuntar a una celda no ocupada por el input. ¿Qué puede decir al respecto si se sabe que  $\lambda \notin L(M)$ ? Justifique su respuesta.**

Se tiene  $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R \rangle$  tal que el cabezal comienza apuntando a la primera celda ocupada por el input.

Se quiere construir  $M' = \langle Q', \Sigma, \Gamma, \delta', q'_0, q'_A, q'_R \rangle$  tal que el cabezal comienza apuntando a cualquier celda de la cinta, inclusive celdas no ocupadas por el input.

Para que  $M$  y  $M'$  sean equivalentes,  $M'$  debe aceptar el mismo lenguaje que  $M$ . Es decir,  $L(M) = L(M')$ . Dicho de otra forma, para toda cadena  $w \in \Sigma^*$ , si  $w \in L(M)$  entonces  $w \in L(M')$ , y si  $w \notin L(M)$  entonces  $w \notin L(M')$ .

Un concepto clave a recordar para resolver este problema, es que las cadenas de entrada nunca contienen el símbolo blanco en ninguna posición, ya que  $B \notin \Sigma$  sea cual sea  $\Sigma$ .

Tenemos dos casos:

1.  $\lambda \notin L(M)$ : En este caso, es posible construir  $M'$  equivalente a  $M$  agregando algunos estados y transiciones que permitan encontrar a la cadena de entrada sin importar dónde comience el cabezal. Tenemos tres casos:

- i. El cabezal comienza en una celda perteneciente a la región de blancos infinita **izquierda**.
  - a.  $M'$  puede moverse hacia la derecha hasta encontrar el primer símbolo distinto de blanco, que es el inicio del input.
  - b. A partir de acá,  $M'$  puede comenzar a ejecutar las mismas transiciones que  $M$  desde ese punto.
- ii. El cabezal comienza en una celda perteneciente a la región de blancos infinita **derecha**.
  - a.  $M'$  puede moverse hacia la izquierda hasta encontrar el primer símbolo distinto de blanco, que es el final del input. Luego, puede seguir moviéndose hacia la izquierda hasta encontrar el primer blanco y luego moverse una celda hacia la derecha y así llegar finalmente al inicio del input.
  - b. A partir de acá,  $M'$  puede comenzar a ejecutar las mismas transiciones que  $M$  desde ese punto.
- iii. El cabezal comienza en una celda ocupada por el input.
  - a. En este caso,  $M'$  no necesita hacer nada especial, ya que el cabezal ya está en una celda ocupada por el input. Simplemente puede comenzar a ejecutar las mismas transiciones que  $M$  desde ese punto.
- iv. Formalmente, los tres casos anteriores se pueden lograr con los siguientes estados y transiciones:
  - a.  $q'_0$ , estado inicial de  $M'$ .
  - b.  $q_{der}$ , estado para moverse hacia la derecha buscando el primer símbolo distinto de blanco.
  - c.  $q_{izq}$ , estado para moverse hacia la izquierda buscando el inicio del input.
  - d.  $q_{principio}$ , estado para ir desde el final del input hasta el inicio del mismo.
  - e.  $\delta'(q'_0, B) = (q_{der}, X, D)$
  - f.  $\delta'(q_{der}, X) = (q_{der}, X, D)$
  - g.  $\delta'(q_{der}, B) = (q_{izq}, X, I)$
  - h.  $\delta'(q_{der}, a) = (q_0, a, S) \forall a \in \Sigma$
  - i.  $\delta'(q_{izq}, X) = (q_{izq}, X, I)$
  - j.  $\delta'(q_{izq}, B) = (q_{der}, X, D)$
  - k.  $\delta'(q_{izq}, a) = (q_{principio}, a, S) \forall a \in \Sigma$
  - l.  $\delta'(q_{principio}, a) = (q_{principio}, a, I) \forall a \in \Sigma$
  - m.  $\delta'(q_{principio}, B) = (q_0, B, D)$
- v. La solución asume que  $X \notin \Sigma$ , es decir, existe al menos un símbolo especial que pertenece a  $\Gamma$  y no a  $\Sigma$ , y además no es el símbolo especial blanco. Este símbolo especial es necesario para que  $M'$  pueda marcar las celdas que ya ha visitado al buscar el inicio del input.

2.  $\lambda \in L(M)$ :

- i. En este caso no es posible construir  $M'$  equivalente a  $M$  que comience con el cabezal apuntando a cualquier celda de la cinta.

- ii. La razón es que si el cabezal comienza en una celda perteneciente a la región de blancos infinita (ya sea izquierda o derecha),  $M'$  nunca podrá encontrar el input, ya que para encontrarlo, la condición necesaria es que el símbolo visto sea distinto de blanco, cosa que jamás ocurrirá si el input es la cadena vacía.
- iii. Es por esto que en este caso,  $M'$  entraría en un loop infinito, y se sabe que cualquier máquina de Turing que entra en un loop infinito dado una cadena de entrada, **rechaza** dicha cadena, no la acepta. Por lo tanto  $\lambda \in L(M)$  pero  $\lambda \notin L(M')$ , y por ende  $L(M) \neq L(M')$ , es decir,  $M$  y  $M'$  no son equivalentes.

## 4. Probar que para toda máquina de Turing $M$ reconocedora del modelo DIS, existe una máquina de Turing $M'$ equivalente con la restricción que no puede cambiar el símbolo de la cinta y mover el cabezal simultáneamente, es decir:

$$\delta'(q_i, a_k) = (q_j, a_l, X) \mid a_k \neq a_l \rightarrow X = S$$

Se tiene  $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R \rangle$  tal que el cabezal comienza apuntando a la primera celda ocupada por el input.

Se quiere construir  $M' = \langle Q', \Sigma, \Gamma, \delta', q'_0, q'_A, q'_R \rangle$  tal que si una transición cambia el símbolo actual de la cinta, entonces el cabezal se queda quieto.

Por ejemplo, si tenemos una transición en  $M'$  como la siguiente:

- $\delta'(q_2, 0) = (q_3, 1, S)$ 
  - En esta transición, el símbolo actual es  $0$  y se cambia a  $1$ , mientras que el cabezal se queda quieto.

Esto se puede lograr añadiendo cuatro transiciones adicionales:

- $\delta'(q_a, a_k) = (q'_{izq}, a_l, S) \mid a_k \neq a_l$ 
  - En esta transición, se cambia el símbolo actual de  $a_k$  a  $a_l$ , y el cabezal se queda quieto. Transicionamos a un estado intermedio para mover el cabezal a la **izquierda**.
- $\delta'(q'_{izq}, a_l) = (q_j, a_l, I)$ 
  - En esta transición, el cabezal se mueve una celda hacia la izquierda y transicionamos al estado al que queríamos ir inicialmente,  $q_j$ .
- $\delta'(q_b, a_k) = (q'_{der}, a_l, S) \mid a_k \neq a_l$

- En esta transición, se cambia el símbolo actual de  $a_k$  a  $a_l$ , y el cabezal se queda quieto.  
Transicionamos a un estado intermedio para mover el cabezal a la **derecha**.
- $\delta'(q'_{der}, a_l) = (q_j, a_l, D)$ 
  - En esta transición, el cabezal se mueve una celda hacia la derecha y transicionamos al estado al que queríamos ir inicialmente,  $q_j$ .

De esta forma, cada vez que en  $M$  haya una transición que cambie el símbolo actual de la cinta y mueva el cabezal, en  $M'$  se reemplaza por estas transiciones que logran el mismo efecto pero respetando la restricción impuesta.

## 5. Probar que para toda máquina de Turing $M$ reconocedora del modelo DIS, existe una máquina de Turing $M'$ equivalente con la restricción que no puede cambiar de estado y mover el cabezal simultáneamente, es decir:

$$\delta'(q_i, a_k) = (q_j, a_l, X) \mid X \in \{D, I\} \rightarrow i = j$$

Se tiene  $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R \rangle$  tal que el cabezal comienza apuntando a la primera celda ocupada por el input.

Se quiere construir  $M' = \langle Q', \Sigma, \Gamma, \delta', q'_0, q'_A, q'_R \rangle$  tal que si una transición mueve el cabezal a izquierda o derecha, entonces el estado no cambia.

Por ejemplo, no podemos tener algo así:

- $\delta'(q_2, 0) = (q_3, 1, D)$ 
  - No se puede, porque el estado cambia de  $q_2$  a  $q_3$  y el cabezal se mueve a la derecha a la vez.

...