Práctica 2

1. Construir máquinas de Turing que:

a. Haga un corrimiento a derecha de la cadena binaria en la cinta, marcando con un símbolo especial $\#$ la celda que corresponde al primer símbolo desplazado. $\Gamma = \{B, \#, 0, 1\}$.

Si tenemos por ejemplo la cadena $w = 0110$ en la cinta, luego de aplicar el corrimiento a derecha, la cinta debería quedar así: $...\#0110...$

```
// Cadena vacía
q0, B
qd, #, D

// Cadena que empieza con 0 → q1
q0, 0
q1, #, D

// Cadena que empieza con 1 → q2
q0, 1
q2, #, D

// Si el segundo símbolo es 1 → q2
q1, 1
q2, 0, D

q1, 0
q1, 0, D

q1, B
qd, 0, D

// Si el segundo símbolo es 0 → q1
q2, 0
q1, 1, D

q2, 1
q2, 1, D

q2, B
qd, 1, D
```

b. Haga un corrimiento a izquierda.

Si tenemos por ejemplo la cadena $w = 0110$ en la cinta, luego de aplicar el corrimiento a izquierda, la cinta debería quedar así: $...0110\#...$

```
// Ir al final de la cadena
q0, 0
q0, 0, D

q0, 1, D
q0, 1, D

q0, B
q1, B, I

// Poner el símbolo # al final de la cadena
q1, 0
q2, #, I

q1, 1
q3, #, I

// Poner ceros
q2, 0
q2, 0, I

q2, 1
q3, 0, I

q2, B
qd, 0, I

// Poner unos
q3, 0
q2, 1, I

q3, 1
q3, 1, I

q3, B
qd, 1, I
```

2. Construir MT:

a. Construir una máquina de Turing M tal que $L(M) = \{0^n 1^n \mid n \geq 1\}$ y mostrar la traza de computación de M para las entradas $w_1 = 0011$ y $w_2 = 011$.

Las cadenas de este lenguaje son aquellas que contienen una cantidad igual de ceros y unos, con todos los ceros antes que los unos. Por ejemplo: $01, 0011, 000111, 00001111$, etc. Claramente $\lambda \notin L$.

La solución consiste en asegurarse de que por cada 0 que se encuentre en la cadena, exista un 1 correspondiente. Por lo tanto se encuentra un 0, se borra, se encuentra un 1, se borra, y se repite el proceso hasta que no queden más 0s. Si al finalizar no quedan 1s, la cadena es aceptada.

```
// Si la cadena es lambda o empieza con 1, se rechaza. Si no, se pasa a q1 y se borra ese primer
q0, B
qR, B, D

q0, 1
qR, 1, D

q0, 0
q1, B, D

// Contamos los ceros. Si viene 1 se rechaza porque no podemos volver a tener unos.
q1, 0
q1, 0, D

q1, B
qR, B, D

q1, 1
q2, 1, D

// Contamos los unos. Si viene 0 se rechaza porque no podemos volver a tener ceros.
q2, 1
q2, 1, D

q2, 0
qR, 0, D

q2, B
q3, B, I

// Borramos el uno de más a la derecha
q3, 1
q4, B, I

// Buscamos el primer blanco o cero
q4, 1
q4, 1, I

q4, 0
q5, 0, I

q4, B
q6, B, D
```

```
// Encontramos el 0 de más a la izquierda
q5, 0
q5, 0, I

q5, B
q0, B, D

// Si es blanco, se borró la misma cantidad de ceros que de unos y la cadena es válida. Si no,
q6, B
qA, B, D

q6, 1
qR, 1, D
```

- Traza para $w_1 = 0011$:
  - $q_0 0011 \vdash Bq_1 011 \vdash B0q_1 11 \vdash B01q_2 1 \vdash B011q_2 B \vdash B01q_3 1B \vdash B0q_4 1BB \vdash Bq_4 01BB \vdash q_5 B01BB \vdash Bq_0 01BB \vdash BBq_1 1BB \vdash BB1q_2 BB \vdash BBq_3 1BB \vdash Bq_4 BBBB \vdash BBq_6 BBB \vdash BBBq_A BB$
- Traza para $w_2 = 011$:
  - $q_0 011 \vdash Bq_1 11 \vdash B1q_2 11 \vdash B11q_2 B \vdash B1q_3 1B \vdash Bq_4 1BB \vdash q_4 B1BB \vdash Bq_6 1BB \vdash B1q_R BB$

b. Construir una máquina de Turing que busque en la cinta el patrón

"abab" y se detenga si y sólo si encuentra ese patrón. $\Gamma = \{a, b, c, B\}$

```
// Buscar la primera 'a'
q0, B
q0, B, D

q0, b
q0, b, D

q0, c
q0, c, D

q0, a
q1, a, D

// Buscar la primera 'b'
q1, B,
q0, B, D

q1, a
q0, a, S

q1, c
q0, c, D

q1, b
q2, b, D

// Buscar la segunda 'a'
q2, B
q0, B, D

q2, b
q0, b, D

q2, c
q0, c, D

q2, a
q3, a, D

// Buscar la segunda 'b'
```

```
q3, B
q0, B, D

q3, a
q0, a, D

q3, c
q0, c, D

q3, b
qA, b, D
```

3. Construir máquinas de Turing para computar las siguientes funciones:

a. Suma unaria. $\Sigma = \{+, 1\}$.

```
// Buscar el signo + y poner 1 en su lugar
q0, B
qd, B, D

q0, 1
q0, 1, D

q0, +
q1, 1, I

// Encontrar el primer 1
q1, 1
q1, 1, I

q1, B
q2, B, D

// Borrar el primer 1
q2, 1
qd, B, D
```

b. Resta unaria $a - b$ con $a > b$. $\Sigma = \{-, 1\}$.

```
// Buscar el signo -
q0, B
qd, B, D

q0, 1
q0, 1, D

q0, -
q1, -, D

// Encontrar el 1 más a la derecha
q1, 1
q1, 1, D

q1, B
q2, B, I

// Borrar el 1 más a la derecha
q2, 1
q3, B, 1

q2, -
qd, B, I

// Ir hasta el signo -
q3, 1
q3, 1, I

q3, -
q4, -, I

// Encontrar el 1 más a la izquierda
q4, 1
q4, 1, I

q4, B
q5, B, D

// Borrar el 1 más a la izquierda
```

```
q5, 1
q0, B, D
```

c. Calcular el complemento a 2 de un número binario de 8 bits. $\Sigma = \{0, 1\}$

```
// Invertir todos los bits
q0, 1
q0, 0, D

q0, 0
q0, 1, D

q0, B
q1, B, I

// Sumar 1, si hace falta se acarrea un uno
q1, 0
qd, 1, I

q1, 1
q2, 0, I

// Sigo sumando ese uno haciendo los corrimientos necesarios
q2, 1
q2, 0, I

q2, 0
qd, 1, I
```

4. Sea $\Sigma = \{a\}$ y $w = a$. Decir cuáles son las palabras que se obtienen como resultado de aplicar las siguientes operaciones:

a. ww

aa

b. www

aaa

c. w^3

$w^3 = www = aaa$

d. w^5

$w^5 = wwwww = aaaaa$

e. w^0

λ

¿Cuáles son sus longitudes?

- $|ww| = 2$
- $|www| = 3$
- $|w^3| = 3$
- $|w^5| = 5$
- $|w^0| = 0$

Definir Σ^*.

$$\Sigma^* = \{\lambda, a, aa, aaa, aaaa, ...\}$$

5. Idem al ejercicio anterior, pero con $\Sigma = \{a, b\}$ y $w = aba$.

- $ww = abaaba$
- $|ww| = 6$
- $www = abaabaaba$
- $|www| = 9$
- $w^3 = www = abaabaaba$
- $|w^3| = 9$

- $w^5 = wwwww = abaabaabaabaaba$
- $|w^5| = 15$
- $w^0 = \lambda$
- $|w^0| = 0$

$$\Sigma^* = \{\lambda, a, b, aa, bb, ab, ba, aba, ...\}$$

6. Sea $\Sigma = \{a, b, c\}$, escriba las 13 cadenas más cortas de Σ^*.

1. λ
2. a
3. b
4. c
5. aa
6. bb
7. cc
8. ab
9. ac
10. ba
11. bc
12. cb
13. ca

7. Dar tres ejemplos de lenguajes basados en el alfabeto $\{0, 1\}$

1. $L_1 = \{\lambda\}$
2. $L_2 = \emptyset$
3. $L_3 = \Sigma^*$

8. ¿Cuántas cadenas de longitud 3 hay en $\{0, 1, 2\}^*$, y cuántas de longitud n?

Tenemos 3 símbolos en el alfabeto, y cada símbolo de una cadena arbitraria w puede tomar cualquiera de ellos 3. Por lo tanto la cantidad de cadenas de longitud 3 que se pueden generar es igual a $3 \times 3 \times 3 = 27$.

Si queremos cadenas de longitud n, se pueden generar $3 \times 3 \times 3 \times \ldots \times 3$ n veces, lo que equivale a 3^n.

9. Explicar la diferencia (si la hay) entre los lenguajes L_1 y L_2.

a. $L_1 = \emptyset, L_2 = \{\lambda\}$

- L_1 es el lenguaje vacío, es decir que no contiene ninguna cadena, ni siquiera la cadena vacía λ.
- L_2 no es un lenguaje vacío, contiene una única cadena que es la cadena vacía λ.
- Por lo tanto $L_1 \neq L_2$.

b. $L_1 = \Sigma^* \cup \{\lambda\}, L_2 = \emptyset \cup \Sigma^*$

- Como $\{\lambda\} \in \Sigma^* \implies \Sigma^* \cup \{\lambda\} = \Sigma^*$
- Como $\emptyset \in \Sigma^* \implies \emptyset \cup \Sigma^* = \Sigma^* \cup \emptyset = \Sigma^*$
- Por lo tanto $L_1 = L_2$.

c. $L_1 = \Sigma^* – \emptyset, L_2 = \Sigma^*$

- Como para todo conjunto, restarle $\emptyset$ resulta en ese mismo conjunto, $\Sigma^* – \emptyset = \Sigma^*$
- Por lo tanto $L_1 = L_2$.

d. $L_1 = \Sigma^* – \{\lambda\}, L_2 = \Sigma^*$

- Como $\{\lambda\} \in \Sigma^* \implies \Sigma^* - \{\lambda\} \neq \Sigma^*$
- Por lo tanto $L_1 \neq L_2$.

10. Mostrar que Σ^* es infinito contable.

Sea Σ un alfabeto finito y no vacío. Entonces, Σ^* es el conjunto de todas las cadenas posibles (incluyendo la cadena vacía) que se pueden formar con los símbolos de Σ.

Σ^* es infinito porque para cualquier cadena de longitud $n \in \mathbb{N}$, siempre se puede formar una cadena de longitud $n+1$ agregando un símbolo adicional del alfabeto Σ. Por lo tanto, no hay un límite superior en la longitud de las cadenas que se pueden formar, lo que implica que Σ^* es infinito.

Σ^* es infinito contable porque se pueden enumerar todos sus elementos creando una correspondencia uno a uno con los números naturales $\mathbb{N}$. Esto se puede lograr mediante el siguiente procedimiento:

1. Se ordenan las cadenas por su longitud: primero la de longitud 0 (la cadena vacía), luego las de longitud 1, luego las de longitud 2, y así sucesivamente.
2. Dentro de cada grupo de cadenas que poseen la misma longitud, se ordenan lexicográficamente (es decir, en orden alfabético).

Por ejemplo, si $\Sigma = \{a, b\}$, la enumeración de Σ^* sería:

1. Longitud 0:
   i. $\lambda \to 0 \in \mathbb{N}$
2. Longitud 1:
   i. $a \to 1 \in \mathbb{N}$
   ii. $b \to 2 \in \mathbb{N}$
3. Longitud 2:
   i. $aa \to 3 \in \mathbb{N}$
   ii. $ab \to 4 \in \mathbb{N}$
   iii. $ba \to 5 \in \mathbb{N}$
   iv. $bb \to 6 \in \mathbb{N}$
4. Longitud 3:
   i. $aaa \to 7 \in \mathbb{N}$
   ii. $aab \to 8 \in \mathbb{N}$
   iii. $aba \to 9 \in \mathbb{N}$
   iv. $abb \to 10 \in \mathbb{N}$
   v. $baa \to 11 \in \mathbb{N}$
   vi. $bab \to 12 \in \mathbb{N}$
   vii. $bba \to 13 \in \mathbb{N}$
   viii. $bbb \to 14 \in \mathbb{N}$
5. $\ldots$

Así, cada cadena en Σ^* puede ser asociada de manera única con un número natural, demostrando que Σ^* es infinito contable.

11. Indicar cuál es el lenguaje que se obtiene al intersectar $(\cap)$ los siguientes lenguajes:

a. $L_1 = \{a^n c^m d^n \mid n \geq 0, m \geq 0\}$ **con** $L_2 = \{c^n \mid n \geq 0\}$

$L_1 \cap L_2 = L_2$

b. $L_1 = \{a^n c^m d^n \mid n > 0, m \geq 0\}$ **con** $L_2 = \{c^n \mid n \geq 0\}$

$L_1 \cap L_2 = \emptyset$

c. $L_1 = \{a^n c^m d^n \mid n \geq 0, m > 10\}$ **con** $L_2 = \{c^n \mid n > 5\}$

$L_1 \cap L_2 = \{c^n \mid n > 10\}$

d. $L_1 = \{1^n 2^m \mid n, m \geq 0,\ \text{n par, m impar}\}$ **con** $L_2 = \{2^n \mid n \geq 0\}$

$\ldots$

e. $L_1 = \{1^n 2^m \mid n, m \geq 0,\ \text{n par, m impar}\}$ **con** $L_2 = \{1^n \mid n \geq 0\}$

$\ldots$

12. Encontrar si es posible un lenguaje L_1 que cumpla:

a. $L_1 \cap \{1^k 2^m 3^n \mid \text{m} = \text{k} + \text{n} + 1 \text{ y n, k} \geq 0\} = \{1^n 2^{n+1} \mid n \geq 0\}$

$\ldots$

b. $L_1 \cap \{1^n 2^m \mid n \neq m \text{ y } n, m \geq 0\} = \{1^n 2^n \mid n > 0\}$

...

13. Conteste las siguientes preguntas sobre Máquinas de Turing

a. ¿Puede el alfabeto de la cinta (Γ) ser el mismo que el alfabeto de entrada (Σ)?

- Es posible que $\Gamma = \Sigma$?
- No, porque siempre se cumple que $B \in \Gamma$ y a la vez $B \notin \Sigma$, para cualquier Σ y Γ.

b. ¿Puede una máquina de Turing tener un único estado?

- Es posible que $|Q| = 1$?
- Sí, porque el único estado obligatorio de una MT es el estado inicial q_0.

c. ¿Cuántos lenguajes existen definidos sobre el alfabeto $\Sigma = \{0, 1\}$? ¿y sobre $\Sigma = \{1\}$?

Cualquier subconjunto de Σ^* es un lenguaje definido sobre Σ. Por lo tanto, la cantidad de lenguajes definidos sobre un alfabeto Σ es igual a la cantidad de subconjuntos de Σ^*, que es infinito. Por lo tanto existen infinitos lenguajes definidos sobre cualquier alfabeto finito no vacío, incluyendo $\Sigma = \{0, 1\}$ y $\Sigma = \{1\}$.

d. ¿Cuáles de los siguientes conjuntos son lenguajes definidos sobre Σ?

i. $\emptyset$

Es un lenguaje porque $\emptyset \subseteq \Sigma^*$. ✅

ii. Σ

Es un lenguaje porque $\Sigma \subseteq \Sigma^*$. ✅

iii. Σ^*

Es un lenguaje porque $\Sigma^* \subseteq \Sigma^*$. ✅

iv. $\{\lambda\}$

Es un lenguaje porque $\{\lambda\} \subseteq \Sigma^*$. ✅

v. $\{\lambda\} \cup \Sigma$

Es un lenguaje porque $\{\lambda\} \cup \Sigma \subseteq \Sigma^*$. ✅

vi. $\{\emptyset\}$

No es un lenguaje porque $\{\emptyset\} \nsubseteq \Sigma^*$. ❌

e. Sea la siguiente máquina de Turing:

$M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R \rangle$ **tal que:**

$Q = \{q_0, q_1, q_2, q_3\}$

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

$\Gamma = \{a, b, c, B\}$

$\delta(q, s) = (q', s', m)$ tal que:

$q \in Q$

$q' \in Q \cup \{q_R\}$

$s, s' \in \Gamma, m \in \{D, I\}$

¿Reconoce el lenguaje $\{\lambda\}$? Si no es así indique cuál es el lenguaje que reconoce.

Como M no posee ninguna transición hacia el estado de aceptación q_A, no es capaz de reconocer ningún lenguaje. Por lo tanto $L(M) = \emptyset$.

14. Sea $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R \rangle$, en cada caso asumir que los $\delta()$ no especificados son los que hacen detener la MT en q_R, determinar $L(M)$:

a.

$Q = \{q_0, q_1\}$

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

$\Gamma = \{0, 1, B\}$

- $\delta(q_0, 0) = (q_0, 0, I)$
- $\delta(q_0, B) = (q_0, B, D)$
- $\delta(q_0, 1) = (q_1, 1, D)$

Como M no tiene ninguna transición hacia el estado de aceptación q_A, no es capaz de reconocer ningún lenguaje. Por lo tanto $L(M) = \emptyset$.

b.

$Q = \{q_0, q_1\}$

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

$\Gamma = \{0, 1, B\}$

- $\delta(q_0, 0) = (q_1, B, D)$
- $\delta(q_1, B) = (q_A, B, D)$
- $\delta(q_1, 0) = (q_A, 0, D)$
- $\delta(q_1, 1) = (q_A, 1, D)$

M reconoce cadenas que empiezan con un 0. Por lo tanto $L(M) = \{0w \mid w \in \Sigma^*\}$.

c.

$$Q = \{q_0, q_1\}$$

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

$$\Gamma = \{0, 1, B\}$$

- $\delta(q_0, 0) = (q_0, 0, I)$
- $\delta(q_0, B) = (q_0, B, D)$
- $\delta(q_0, 1) = (q_1, 1, D)$
- $\delta(q_1, 0) = (q_0, B, I)$
- $\delta(q_1, 1) = (q_0, B, D)$

Nuevamente M no tiene ninguna transición hacia el estado de aceptación q_A, por lo tanto no es capaz de reconocer ningún lenguaje. $L(M) = \emptyset$.

d.

$$Q = \{q_0\}$$

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

$$\Gamma = \{0, 1, B\}$$

- $\delta(q_0, 1) = (q_0, B, I)$
- $\delta(q_0, 0) = (q_A, B, I)$
- $\delta(q_0, B) = (q_0, B, D)$

M reconoce cadenas que contienen al menos un 0. Por lo tanto $L(M) = \{w \mid \text{w contiene al menos un cero}\}$.

e.

$$Q = \{q_0, q_1\}$$

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

$$\Gamma = \{0, 1, B\}$$

- $\delta(q_0, 0) = (q_1, B, D)$
- $\delta(q_1, 0) = (q_1, 1, D)$
- $\delta(q_1, 1) = (q_1, 0, D)$
- $\delta(q_1, B) = (q_A, 1, D)$

M reconoce cadenas que empiezan con un 0. Por lo tanto $L(M) = \{0w \mid w \in \Sigma^*\}$.