

Ejercicios sobre Máquinas de Turing

LENGUAJES Y AUTOMATAS 801 ISC TRAN

David Santiago Castillo Molano

Docente

Ing. FABIO ALEJANDRO SASTOQUE RINCON

Universidad de Cundinamarca

Seccional Ubaté

**Ejercicios sobre maquinas de Turing**

**Lenguajes y autómatas octavo semestre**

**Ingeniería de sistemas y computación**

**Universidad de cundinamarca Seccional, Ubaté**

**Problema 1**

Máquina de Turing que proporciona el complemento a 1 de un número binario.

**Problema 2**

Diseñar una máquina de Turing que calcula el número consecutivo de un número dado en binario.

**Problema 3**

Diseñar una máquina de Turing que acepta el lenguaje

$$L = \{0^n 1^n : n > 0\}$$

**Problema 4**

Diseñar una Máquina de Turing que, dada una palabra  $w$  del alfabeto  $\Sigma = \{0, 1\}$ , proporciona su reverso,  $w_R$ .

# DESARROLLO

## Problema 1

Máquina de Turing que proporciona el complemento a 1 de un número binario.

```
C:\> Users > santo > Documents > UDEC > 2025-2 > LENGUAJES Y AUTOMATAS > 4 ejercicios maquina de turing > Problema_1.py > ...
1  def Problema_1(cadena_entrada):
2      # Configuración inicial
3      cinta = list(cadena_entrada)
4      cabezal = 0
5      estado = 'q0'
6
7      # Tabla de transiciones: (estado_actual, caracter_leido) -> (nuevo_estado, escribir, movimiento)
8      transiciones = {
9          ('q0', '0'): ('q0', '1', 'R'), # Lee 0, escribe 1, mueve Derecha
10         ('q0', '1'): ('q0', '0', 'R'), # Lee 1, escribe 0, mueve Derecha
11         ('q0', 'B'): ('halt', 'B', 'L') # Fin de la cinta
12     }
13
14     pasos = 0
15     while estado != 'halt' and pasos < 1000:
16         # Manejo de cinta infinita
17         if cabezal < 0: cinta.insert(0, 'B'); cabezal = 0
18         if cabezal >= len(cinta): cinta.append('B')
19
20         caracter = cinta[cabezal]
21         clave = (estado, caracter)
22
23         if clave in transiciones:
24             nuevo_estado, escribir, direccion = transiciones[clave]
25             cinta[cabezal] = escribir
26             estado = nuevo_estado
27             cabezal += 1 if direccion == 'R' else -1
28         else:
29             break # Detener si no hay transición definida
30         pasos += 1
31
32     return "".join(cinta).replace('B', '')
33
34     print(f"Complemento de '10110': {Problema_1('10110')}")
```

```
PS C:\Users\santo> & C:/Users/santo/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/santo/Documents/UDEC/2025-2/LENGUAJES Y AUTOMATAS/4 ejercicios maquina de turing/Problema_1.py"
Complemento de '10110': 01001
PS C:\Users\santo> & C:/Users/santo/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/santo/Documents/UDEC/2025-2/LENGUAJES Y AUTOMATAS/4 ejercicios maquina de turing/Problema_1.py"
Complemento de '10110': 01001
PS C:\Users\santo>
```

## Problema 2

Diseñar una máquina de Turing que calcula el número consecutivo de un número dado en binario.

```
C:\Users\santo> Documents > UDEC > 2025-2 > LENGUAJES Y AUTOMATAS > 4 ejercicios maquina de turing > Problema_2.py > Problema_2
1 def Problema_2(cadena_entrada):
2     cinta = list(cadena_entrada)
3     cabezal = 0
4     estado = 'q0'
5
6     transiciones = {
7         # Fase 1: Ir al final de la derecha
8         ('q0', '0'): ('q0', '0', 'R'),
9         ('q0', '1'): ('q0', '1', 'R'),
10        ('q0', 'B'): ('q1', 'B', 'L'), # Se Encontró el final, cambia a modo suma (q1)
11
12        # Fase 2: Sumar 1 moviéndose a la izquierda
13        ('q1', '0'): ('halt', '1', 'R'), # 0->1, no hay acarreo, fin.
14        ('q1', '1'): ('q1', '0', 'L'), # 1->0, llevamos acarreo a la izquierda.
15        ('q1', 'B'): ('halt', '1', 'R')
16    }
17
18    pasos = 0
19    while estado != 'halt' and pasos < 1000:
20        if cabezal < 0: cinta.insert(0, 'B'); cabezal = 0
21        if cabezal >= len(cinta): cinta.append('B')
22
23        caracter = cinta[cabezal]
24        clave = (estado, caracter)
25
26        if clave in transiciones:
27            nuevo_estado, escribir, direccion = transiciones[clave]
28            cinta[cabezal] = escribir
29            estado = nuevo_estado
30            cabezal += 1 if direccion == 'R' else -1
31        else:
32            break
33        pasos += 1
34
35    return "".join(cinta).replace('B', '')
36
37
38    print(f"Sucesor de '1011': {Problema_2('1011')}")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\santo> & C:/Users/santo/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/santo/Documents/UDEC/2025-2/LENGUAJES Y AUTOMATAS/4 ejercicios maquina de
Sucesor de '1011': 1100
PS C:\Users\santo>
```

### Problema 3

Diseñar una máquina de Turing que acepte el lenguaje

$$L = \{0^n 1^n : n > 0\}$$

```
C:\Users\santo> Documents > UDEC > 2025-2 > LENGUAJES Y AUTOMATAS > 4 ejercicios maquina de turing > Problema_3.py > Problema_3
1 def Problema_3(cadena_entrada):
2     cinta = list(cadena_entrada)
3     cabezal = 0
4     estado = 'q0'
5
6     transiciones = {
7         # q0: Buscar un 0 para marcar con X
8         ('q0', '0'): ('q1', 'X', 'R'),
9         ('q0', 'Y'): ('q3', 'Y', 'R'), # Si encontramos Y al inicio, verificar si terminamos
10
11         # q1: Avanzar a la derecha buscando el 1 correspondiente
12         ('q1', '0'): ('q1', '0', 'R'),
13         ('q1', 'Y'): ('q1', 'Y', 'R'),
14         ('q1', '1'): ('q2', 'Y', 'L'), # Encontró el 1, lo marca con Y y regresa
15
16         # q2: Regresar a la izquierda buscando la X
17         ('q2', '0'): ('q2', '0', 'L'),
18         ('q2', 'Y'): ('q2', 'Y', 'L'),
19         ('q2', 'X'): ('q0', 'X', 'R'), # Encontró X, reinicia el ciclo
20
21         # q3: Verificación final (asegurar que no queden caracteres sin emparejar)
22         ('q3', 'Y'): ('q3', 'Y', 'R'),
23         ('q3', 'B'): ('accept', 'B', 'L') # Éxito
24     }
25
26     pasos = 0
27     while pasos < 2000:
28         if estado == 'accept': return True
29
30         if cabezal < 0: cinta.insert(0, 'B'); cabezal = 0
31         if cabezal >= len(cinta): cinta.append('B')
32
33         caracter = cinta[cabezal]
34         clave = (estado, caracter)
35
36         if clave in transiciones:
37             nuevo_estado, escribir, direccion = transiciones[clave]
38             cinta[cabezal] = escribir
39             estado = nuevo_estado
40             cabezal += 1 if direccion == 'R' else -1
41         else:
42             break # Rechazo implícito
43     pasos += 1
```

```
43         pasos += 1
44
45     return estado == 'accept'
46
47
48     print(f"¿Es '0011' válido?: {Problema_3('0011')}")
49     print(f"¿Es '001' válido?: {Problema_3('001')}")
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
¿Es '0011' válido?: True
¿Es '001' válido?: False
PS C:\Users\santo>
```

## Problema 4

Diseñar una Máquina de Turing que, dada una palabra  $w$  del alfabeto  $\Sigma=\{0, 1\}$ , proporciona su reverso,  $w_R$ .

```
C:\Users\santo > Documents > UDEC > 2025-2 > LENGUAJES Y AUTOMATAS > 4 ejercicios maquina de turing > Problema_4.py > Problema_4

1 def Problema_4(cadena_entrada):
2     cinta = list(cadena_entrada)
3     cabezal = 0
4     estado = 'q0'
5
6     transiciones = {
7         # q0: Ir al final de la parte "sin procesar"
8         ('q0', '0'): ('q0', '0', 'R'),
9         ('q0', '1'): ('q0', '1', 'R'),
10        ('q0', 'X'): ('q_check', 'X', 'L'), # X/Y marcan bits ya procesados
11        ('q0', 'Y'): ('q_check', 'Y', 'L'),
12        ('q0', '#'): ('q_check', '#', 'L'), # Separador (si existiera)
13        ('q0', 'B'): ('q_check', 'B', 'L'),
14
15        # q_check: Mirar qué bit debemos mover
16        ('q_check', '0'): ('q_carry0', 'X', 'R'), # Es un 0: mácalo X y llévalo
17        ('q_check', '1'): ('q_carry1', 'Y', 'R'), # Es un 1: mácalo Y y llévalo
18        ('q_check', 'X'): ('q_clean', 'X', 'R'), # Solo quedan marcas: ir a limpiar
19        ('q_check', 'Y'): ('q_clean', 'Y', 'R'),
20        ('q_check', 'B'): ('q_clean', 'B', 'R'), # Cinta vacía o fin
21
22        # q_carry0: Llevar el 0 al final de la cinta
23        ('q_carry0', 'X'): ('q_carry0', 'X', 'R'),
24        ('q_carry0', 'Y'): ('q_carry0', 'Y', 'R'),
25        ('q_carry0', '#'): ('q_write0', '#', 'R'),
26        ('q_carry0', 'B'): ('q_write0', '#', 'R'), # Crea separador # virtualmente
27        ('q_carry0', '0'): ('q_carry0', '0', 'R'),
28        ('q_carry0', '1'): ('q_carry0', '1', 'R'),
29
30        ('q_write0', '0'): ('q_write0', '0', 'R'),
31        ('q_write0', '1'): ('q_write0', '1', 'R'),
32        ('q_write0', 'B'): ('q_return', '0', 'L'), # Escribe el 0
33
34        # q_carry1: Llevar el 1 al final de la cinta
35        ('q_carry1', 'X'): ('q_carry1', 'X', 'R'),
36        ('q_carry1', 'Y'): ('q_carry1', 'Y', 'R'),
37        ('q_carry1', '#'): ('q_writel', '#', 'R'),
38        ('q_carry1', 'B'): ('q_writel', '#', 'R'),
39        ('q_carry1', '0'): ('q_carry1', '0', 'R'),
40        ('q_carry1', '1'): ('q_carry1', '1', 'R'),
41
42        ('q_writel', '0'): ('q_writel', '0', 'R'),
43        ('q_writel', '1'): ('q_writel', '1', 'R'),
44        ('q_writel', 'B'): ('q_return', '1', 'L'), # Escribe el 1
45    }
```

```

46     # q_return: Volver al inicio absoluto para reiniciar el ciclo
47     ('q_return', '0'): ('q_return', '0', 'L'),
48     ('q_return', '1'): ('q_return', '1', 'L'),
49     ('q_return', '#'): ('q_return', '#', 'L'),
50     ('q_return', 'X'): ('q_return', 'X', 'L'),
51     ('q_return', 'Y'): ('q_return', 'Y', 'L'),
52     ('q_return', 'B'): ('q0', 'B', 'R'),
53
54     # q_clean: Borrar la basura (X, Y, #) de la izquierda
55     ('q_clean', 'X'): ('q_clean', 'B', 'R'),
56     ('q_clean', 'Y'): ('q_clean', 'B', 'R'),
57     ('q_clean', '#'): ('q_clean', 'B', 'R'),
58     ('q_clean', '0'): ('halt', '0', 'L'), # Encontró el resultado limpio
59     ('q_clean', '1'): ('halt', '1', 'L'),
60     ('q_clean', 'B'): ('halt', 'B', 'L')
61 }
62
63 pasos = 0
64 while estado != 'halt' and pasos < 5000:
65     if cabezal < 0: cinta.insert(0, 'B'); cabezal = 0
66     if cabezal >= len(cinta): cinta.append('B')
67
68     caracter = cinta[cabezal]
69     clave = (estado, caracter)
70
71     if clave in transiciones:
72         nuevo_estado, escribir, direccion = transiciones[clave]
73         cinta[cabezal] = escribir
74         estado = nuevo_estado
75         cabezal += 1 if direccion == 'R' else -1
76     else:
77         break
78     pasos += 1
79
80     return "".join(cinta).replace('B', '')
81
82 # Prueba
83 print(f"Reverso de '001': {Problema_4('001')}")

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\santo> & C:/Users/santo/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/santo/Documents/UDEC/2025-2/LENGUAJES Y AUTOMATAS/4 ejercicios maquina de turing/Problema_4.py"
Reverso de '001': 100
PS C:\Users\santo>

```

**Se anexa repositorio:**

[https://github.com/David-Santiago-Castillo/4\\_Ejercicios\\_turing.git](https://github.com/David-Santiago-Castillo/4_Ejercicios_turing.git)