INSTITUTO POLITECNICO NACIONAL

Escuela Superior de Cómputo

PRACTICA 9

Autor:
Gomez Hernandez Alan Javier

27 diciembre, 2021|

MAQUINA TURING

RESUMEN

Para este programa se desarrollo una maquina de Turing que acepte el lenguaje $0^n1^n|n >= 1$. **Que de manera formal se define como lo siguiente:** $M = (q0, q1, q2, q3, q4, 0, 1, 0, 1, X, Y, B, \delta, q0, B, q4)$ **donde** δ **se especifica como en la siguiente tabla:** El funcionamiento de esta

Símbolo					
Estado	0	1	X	Y	B
q_0	(q_0, X, R)	-	-	(q_0, X, R)	-
q_1	(q_0, X, R)	(q_0, X, R)	-	(q_0, X, R)	-
q_2	(q_0, X, R)	-	(q_0, X, R)	(q_0, X, R)	-
q_3	-	-	-	(q_0, X, R)	(q_0, X, R)
q_4	-	-	-	-	-

Figure 1: Tabla 1

maquina se puede entender mejor con el diagrama de la figura

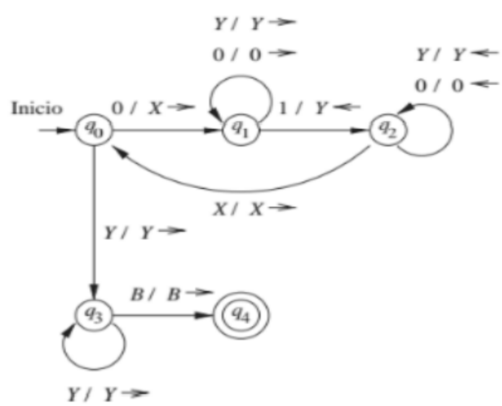

Figure 2:

Diagrama de transiciones de una Maquina de Turing que acepta las cadenas 0^n1^n.[1] El programa debe de contar con un modo manual y automático, en ambos modos se ingresara una cadena de ceros y unos que sera trabajada por la maquina de Turing a través de una unidad de control y una banda (véase la figura 3) la cual sera la cadena que se ingrese y se mostrara si es una cadena valida junto al proceso que se realizo para llegar a dicho resultado.

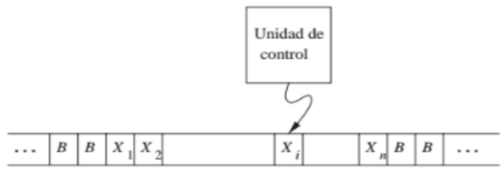

Figure 3:

INTRODUCCION

La máquina de Turing, presentada por Alan Turing en 1936 en On computable numbers, with an application to the Entscheidungsproblems, es el modelo matemático de un dispositivo que se comporta como un autómata finito y que dispone de una cinta de longitud infinita en la que se pueden leer, escribir o borrar símbolos. Existen otras versiones con varias cintas, deterministas o no, etc., pero todas son equivalentes (respecto a los lenguajes que aceptan). Uno de los teoremas más importantes sobre las máquinas de Turing es que pueden simular el comportamiento de una computadora (almacenamiento y unidad de control). Por ello, si un problema no puede ser resuelto por una de estas máquinas, entonces tampoco puede ser resuelto por una computadora (problema indecidible, NP). La notación de las máquinas de Turing es sencilla y exacta, por lo que es más cómodo trabajar con ellas a la hora de estudiar qué problemas son decidibles (P) y cuáles indecidibles (NP). Llamamos Máquina de Turing (ó MT) a $M = (q0, q1, q2, q3, q4, 0, 1, 0, 1, X, Y, B, \delta, q0, B, q4)$ donde

- Q es el conjunto finito de estados que denotaremos por q0,q1,q2,.....
- $\sum$ es el alfabeto: el conjunto finito de símbolos de entrada.
- T es el conjunto de símbolos de cinta. El alfabeto es un subconjunto de T.
- q0 es el estado inicial: el estado en el que se encuentra inicialmente la MT.
- B es un elemento de
$$\sum$$
: el símbolo en blanco. Se encuentra en todas las casillas de la cinta que no tienen un símbolo de entrada.
- F es el conjunto de estados finales.
- δ es la función de transiciones.

La expresión $\delta(q, X) = (p, Y, D)$ indica que en el estado q, si la cabeza de la MT señala al símbolo de cinta X, entonces la MT escribe el símbolo de cinta Y en la casilla actual (cambia X por Y) y mueve la cabeza una casilla hacia D (D puede ser derecha, R; o izquierda, L) y pasa al estado p. La cinta de la MT está formada por infinitas casillas. Inicialmente, la palabra de entrada (una concatenación de símbolos del alfabeto) se encuentra escrita en casillas consecutivas de la cinta y la cabeza señala al primer símbolo de la palabra. Todas las otras casillas (hacia la izquierda y la derecha) contienen el símbolo en blanco.

DESARROLLO

0.1 metodos (CODIGO)

```
# -*- coding: utf-8 -*-
from __future__ import print_function

def maquina(cadena):
    continuar = True
    archivo = open('turing-historia.txt', 'w')
    i = 0
    estado = 0
    cadena_aux = list(cadena)
    cadena_final = ''
    archivo.write('La cadena es: %s \n' %cadena)
    while continuar:
        try:
            simbolo = cadena_aux[i]
        except Exception as e:
            cadena_aux.append('B')
            simbolo = cadena_aux[i]
        cadena_final = imprimir_secuencia(estado, cadena_aux, i)
        print(cadena_final, end = '')
        archivo.write(cadena_final)
        resultado = funcion_transicion(estado, simbolo)
        if len(resultado) == 0:
            break
        estado = resultado[0]
        cadena_aux[i] = resultado[1]
        if resultado[2] == 'R':
            i += 1
        elif resultado[2] == 'L':
            i -= 1
        print(' |- ', end='')
        archivo.write(' |- ')
    print('\n')
    if estado == 4:
        print('Cadena valida')
        archivo.write('\n\nCadena valida')
    else:
        print('Cadena invalida')
        archivo.write('\nCadena invalida')
    archivo.close()

def imprimir_secuencia(estado, cadena, indice):
    cadena_aux = ''
```

```
    i = 0
    while i<len(cadena):
        if i == indice:
            cadena_aux += '(q'+ str(estado) + ')'
        cadena_aux += cadena[i]
        i +=1
    return cadena_aux

def funcion_transicion(estado, simbolo):
    if estado == 0:
        return estado_cero(simbolo)
    elif estado == 1:
        return estado_uno(simbolo)
    elif estado == 2:
        return estado_dos(simbolo)
    elif estado == 3:
        return estado_tres(simbolo)
    elif estado == 4:
        return estado_cuatro(simbolo)

def estado_cero(simbolo):
    if simbolo == '0':
        return [1, 'X', 'R']
    elif simbolo == 'Y':
        return [3, 'Y', 'R']
    return []

def estado_uno(simbolo):
    if simbolo == '0':
        return [1, '0', 'R']
    elif simbolo == '1':
        return [2, 'Y', 'L']
    elif simbolo == 'Y':
        return [1, 'Y', 'R']
    return []

def estado_dos(simbolo):
    if simbolo == '0':
        return [2, '0', 'L']
    elif simbolo == 'X':
        return [0, 'X', 'R']
    elif simbolo == 'Y':
        return [2, 'Y', 'L']
    return []

def estado_tres(simbolo):
```

```
    if simbolo == 'Y':
        return [3, 'Y', 'R']
    elif simbolo == 'B':
        return [4, 'B', 'R']
    return []

def estado_cuatro(simbolo):
    return []
```

0.2 main-turing (CODIGO)

```
# -*- coding: utf-8 -*-
from __future__ import print_function
from metodos import maquina
import random

separador = '='*50
def iniciar():
    continuar = True
    while continuar:
        opcion = imprimir_menu()
        if opcion == 1:
            entrada_consola()
        elif opcion == 2:
            ejecutar_random()
        else:
            break # Sal del programa
        print('=' * 100)
        opcion = input("Reintentar [s/n]: ")
        if opcion.lower() != 's':
            continuar = False

    print('Saliendo del programa...')

def imprimir_menu():
    print('\n\n%sMenu%s' % (separador, separador))
    print("""
        1.- Entrada en consola (Manual)
        2.- Aleatorio (Automatico)
        3.- Salir
    """)
    try:
        opcion = int(input("Selecciona una opcion valida: "))
        return opcion
    except Exception as e:
```

```
        print('Error ', e)
        return 0

def entrada_consola():
    texto = input("Escribe un numero binario: ")
    print('\n Historia de la maquina de Turing')
    maquina(texto)

def ejecutar_random():
    i = 0
    longitud_random = random.randint(1, 1000)
    binario = ''
    while i < longitud_random:
        binario += random.choice(['0', '1'])
        i += 1

    print("La cadena es: ", binario)
    print('\n Historia de la maquina de Turing')
    maquina(binario)

iniciar()
```

0.3 FUNCIONAMIENTO Y PRUEBAS

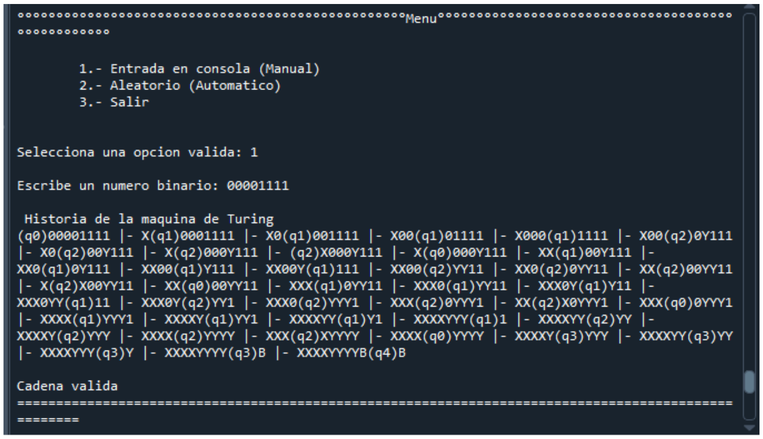

Figure 4: Menú Manua, donde ingresamos una cadena validal

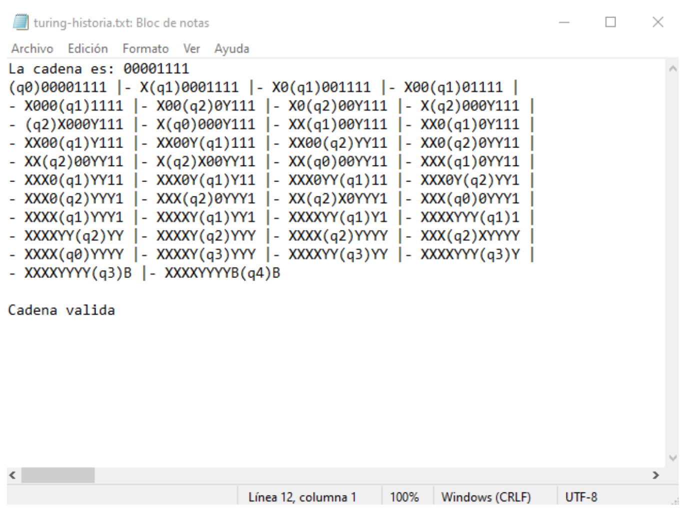

Figure 5: Historia del automata maquina turing, de la cadena valida

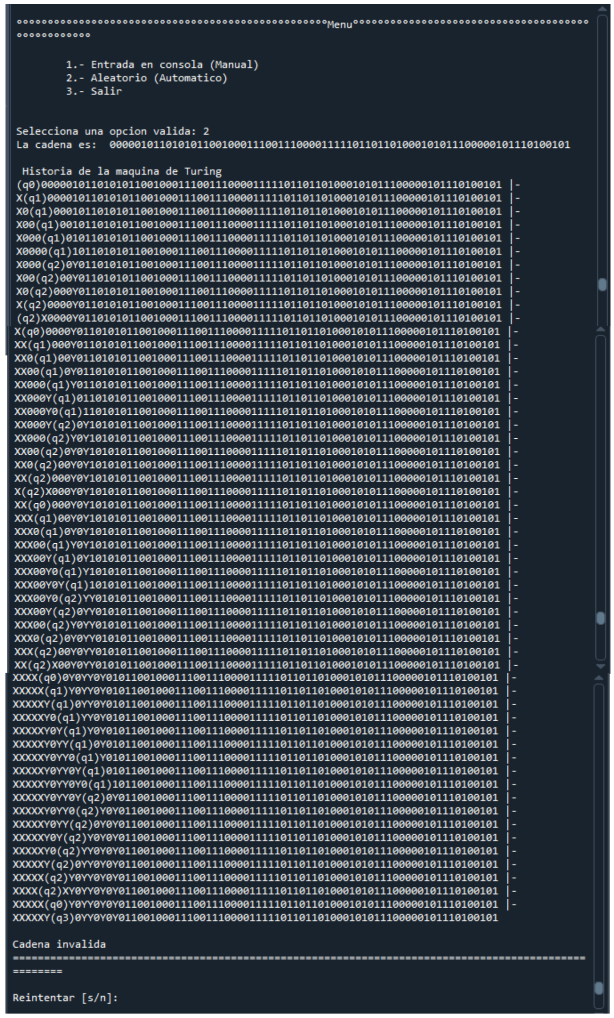

Figure 6: Menu autómatico

1 CONCLUSION

Las máquinas de Turing son elementales no solo en la resolución de problemas indefinibles o intratables; ya que estos ayudan a los ingenieros en programación a usarlos como una metodología de análisis y diseño en la resolución de dicho problema. Las máquinas de Turing han venido a facilitar y ayudar al programador como al ser humano en las tomas de decisiones en cuanto a problemas de la vida cotidiana. Ya que en casos particulares se tiende a llegar al desbordamiento de ideas en cuanto al análisis y diseño de respuestas a un suceso en particular de la vida diaria, además de ello las máquinas de Turing ayudan en la estandarización de las máquinas electrónicas para poder entrar de un estado a otro; es decir, que estas pueden llegar a pensar que decisión tomar, cuando estén sometidas o se encuentren en una situación dada en un contexto en especial. Esto de manera general, ahora siendo mas precisos para la resolucion de la maquina turing podemos comprender el principio basico de esta y como funciona de manera individual, siendo similar a una arquitectura de computadora segun alguna cadena ingresada valida o no valida segun el caso que se realizo.

References

[1] J. E. Hopcroft, R. Motwani, and J. D. Ullman, Introduccion a La Teorıa De Automatas, Lenguajes Y Computacion. Addison-Wesley, 2007.

[2] J. D. Ullman, "Finite Automata." http://infolab.stanford.edu/ ullman/ialc/spr10/slides/fa1.pdf, 2010. [Consultado: 2021-12-20].

[3] M. (2014d, noviembre 11). Máquina de Turing: teoría de la computación: lenguaje, ejemplos y teoremas. Maquina Turing. Recuperado 26 de diciembre de 2021, de https://www.matesfacil.com/automatas-lenguajes/Maquina-Turing.html