

Instituto Politécnico Nacional

ESCUELA SUPERIOR DE COMPUTACIÓN

MAQUINA DE TURING

Práctica 7

Alumno:

Alcántara Covarrubias Erik

Profesor:

Juarez Martinez Genaro

Grupo: 4CM6

1 Introducción

En términos generales podemos decir que una máquina de Turing es un dispositivo capaz de procesar cualquier clase de datos, codificados mediante palabras o cadenas formadas por símbolos dados. Esta máquina abstracta consta de una cinta infinita hacia ambos lados que está dividida en casillas o celdas, las cuales contienen a lo más un símbolo; también tiene una cabeza de lectura y escritura que se mueve a lo largo de la cinta. Su funcionamiento es secuencial y está dado por la llamada función de transición que se representa mediante una tabla finita de reglas, llamadas de ejecución o transición, que describen la acción que debe ejecutarse considerando el estado actual y el símbolo escrito en la celda donde se encuentra la cabeza, con lo cual se determinan el nuevo estado, el símbolo que se debe escribir y el desplazamiento de la cabeza hacia alguna de las celdas vecinas.

Al principio, la máquina se encuentra en un estado inicial fijo, la cadena o dato de entrada se asume escrito en algún lugar de la cinta y la cabeza se encuentra en una celda a la izquierda del primer símbolo de esta. Cabe resaltar que en todas las celdas de la cinta, excepto aquellas que contienen la cadena, se encuentra escrito el símbolo especial llamado blanco, por lo tanto, al inicio de la ejecución la cabeza estará leyendo un blanco.

La ejecución es dirigida por la función de transición y termina cuando no se ha definido ninguna regla para el estado y símbolo actual, es decir cuando no es posible ejecutar una acción más. Decimos que una cadena es aceptada por la máquina cuando la ejecución de esta termina y se encuentra en el estado distinguido como final. Adicionalmente la cadena final escrita sobre la máquina puede considerarse como la respuesta o dato de salida del proceso.

2 Marco teórico

La máquina consta de una unidad de control, que puede encontrarse en cualquiera de un conjunto finito de estados. Hay una cinta dividida en cuadrados o casillas y cada casilla puede contener un símbolo de entre un número finito de símbolos.

La notación formal que vamos a emplear para una máquina de Turing es similar a la que hemos empleado para los autómatas finitos o los autómatas a pila. Describimos una máquina de Turing mediante la siguiente séptula:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

cuyos componentes tienen el siguiente significado:

- Q : El conjunto finito de estados de la unidad de control.
- Σ : El conjunto finito de símbolos de entrada.
- Γ : El conjunto completo de símbolos de cinta: Σ siempre es un subconjunto de Γ .
- δ : La función de transición. Los argumentos de $\delta(q, X)$ son un estado de q y un símbolo de cinta X . El valor de $\delta(q, X)$, si está definido, es (p, Y, D) , donde:
 1. p es el siguiente estado de Q .
 2. Y es el símbolo de Γ , que se escribe en la casilla que señala la cabeza y que sustituye a cualquier símbolo que se encontrara en ella.
 3. D es una dirección y puede ser L o R , lo que nos indica la dirección en que la cabeza se mueve, "izquierda" (L) o "derecha" (R), respectivamente.
- q_0 : El estado inicial, un elemento de Q , en el que inicialmente se encuentra la unidad de control.
- B : El símbolo espacio en blanco. Este símbolo pertenece a Γ pero no a Σ ; es decir, no es un símbolo de entrada. El espacio en blanco aparece inicialmente en todas las casillas excepto en aquellas que se almacenan los símbolos de la entrada.
- F : El conjunto de estados finales o de aceptación, un subconjunto de Q .

3 Explicación del problema e implementación del algoritmo

3.1 Problema a resolver

Se programará la máquina del artículo titulado "What can we learn from universal Turing machines?".

1. La máquina se tiene que animar para cadenas pequeñas (menor igual a 10 caracteres).
2. Puede recibir la cadena por parte del usuario o aleatoriamente con un máximo de 50 caracteres.
3. Mandar la salida a u archivo de texto que muestre las descripciones instantáneas (IDs) por renglón en cada iteración.

3.2 Implementación

```
import random
import os
from time import sleep

def Maquina (cad, anim):
    """
    It takes a string and a boolean as input, and returns a string

    :param cad: The string to be processed
    :param anim: True or False, if you want to see the animation or not
    """
    estado=1
    j=0
    archivo = open(" Practica7 /MT.txt", "w+")
    while (True):
        try:
            archivo.write(' '.join(cad) + " " + str(estado))
            archivo.write("\n")
            if (anim):
                animacion(' '.join(cad), j)
            if (estado == 1):
                if (cad[j] == '*'):
                    cad[j] = 'X'
                    j+=1
                    estado = 2
                continue
            elif (estado == 2):
                if (cad[j]=='*'):
                    j+=1
                    estado= 3
                elif (cad[j]=='|'):
                    j+=1
                    estado =2
                continue
            elif estado == 3:
                if (cad[j]=='*'):
                    cad[j] = 'X'
                    j-=1
                    estado = 4
                elif (cad[j]=='|'):
                    j+=1
                    estado= 3
                continue
```

```

elif estado == 4:
    if (cad[j]=='*'):
        j-=1
        estado= 4
    elif (cad[j]=='|'):
        cad[j] = 'a'
        j+=1
        estado=5
    elif (cad[j]=='X'):
        j+=1
        estado=7
    continue
elif estado == 5:
    if (j == (len(cad)-1)):
        cad.append('|')
        j-=1
        estado=6
    elif (cad[j]=='*'):
        j+=1
        estado=5
    elif (cad[j]=='|'):
        j+=1
        estado=5
    elif (cad[j]=='X'):
        j+=1
        estado = 5
    continue
elif estado == 6:
    if (cad[j]=='*'):
        j-=1
        estado=6
    elif (cad[j]=='|'):
        j-=1
        estado=6
    elif (cad[j]=='a'):
        cad[j]='|'
        j-=1
        estado=4
    elif (cad[j]=='X'):
        j-=1
        estado=6
    continue
elif estado == 7:
    if (cad[j]=='*'):
        j+=1
        estado=8
    elif (cad[j]=='|'):
        j+=1
        estado=7
    continue
elif estado == 8:
    if (j == (len(cad)-1)):
        cad.append('*')
        j-=1
        estado=9
    elif (cad[j]=='|'):

```

```

        j+=1
        estado=8
    elif (cad[j]=='X'):
        cad[j]='*'
        j+=1
        estado=8
    continue
elif estado == 9:
    if (cad[j]=='*'):
        j-=1
        estado=9
    elif (cad[j]=='|'):
        j-=1
        estado=9
    elif (cad[j]=='X'):
        cad[j]='*'
        estado=9
        archivo.write(''.join(cad))
        if (anim):
            animacion(''.join(cad), j)
        break
    continue
else:
    print("Error")
    break

except IndexError:
    break

print("La cadena final es: {}".format(''.join(cad)))

def animacion(cadena, posicion):
    """
    It prints the string, then prints a caret at the position specified.

    :param cadena: The string to be animated
    :param posicion: The position of the caret
    """
    animac = []
    if os.name == "nt":
        os.system("cls")
    else:
        os.system("clear")
    print(cadena)
    for i in range(posicion-1):
        animac.append(" ")
    animac.append("^")
    print(''.join(animac))
    sleep(0.09)

if __name__ == "__main__":

    while True:

        print("\n*****PROGRAMA 7: MAQUINA DE TURING*****")
        print("1.-Manual")

```

```

print("2.- Automatico")
print("3.- Salir")
OP = int(input("Elija una opcion: "))
if OP==1:
    n = int( input("Cual es el valor de n: "))
    m = int(input("Cual es el valor de m: "))

elif OP==2:
    n = random.randint(1, 50)
    m = random.randint(1, 50)

elif OP==3:
    print("Adios!!\n")
    exit()

else:
    print("No existe esa opcion\n")
    break

cad = []
aux = []
aux_2 = []
for i in range (n):
    aux.append('|')
for i in range (m):
    aux_2.append('|')

cad = list('*'+ ''.join(aux) + '*'+ ''.join(aux_2) + '*')
print("Cadena original: "+ ''.join(cad))
largo = len(cad)
if largo <= 10:
    animaniac = True
else:
    print("No se puede realizar la animacion\n")
    animaniac = False
Maquina(cad, animaniac)

```

3.3 Capturas de Resultados

4 Conclusión

5 Bibliografía

- Introduction to Automata Theory, Languages, and Computation John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman Pearson/Addison Wesley, 2nd edition
- Software Foundation, P. (2022, 13 enero). 3.10.2 Documentation. Documentación de Python. Recuperado 15 de febrero de 2022, de <https://docs.python.org/es/3/>