



Universidad Nacional Autónoma De México

FACULTAD DE INGENIERÍA

LENGUAJES FORMALES Y AUTÓMATAS

Proyecto Final

Grupo: 4

Máquina de Turing

Profesora:

LUCILA PATRICIA ARELLANO MENDOZA

Integrantes:

Aguilar Martinez Erick Yair

Ceniceros Mariaca Carlos

López Ramírez Monserrat



Índice

1. Introducción	3
2. Análisis:	4
3. Diseño	5
4. Funcionalidad:	9
5. Conclusiones	15



Lenguajes Formales y Autómatas

Máquina de Turing

Resumen

El presente proyecto expone el funcionamiento de una máquina de Turing de doble cinta que realiza la función de multiplicación, se aborda la definición así como el diseño del Automata correspondiente y su implementación en Lenguaje C.

1. Introducción

Las máquinas de Turing, lejos de limitarse al reconocimiento de lenguajes, desempeñan un papel fundamental como modelos de la computación. La máquina de Turing fue creada por Alan Turing, representan la base conceptual de los dispositivos computacionales. La idea consiste es que al construir una serie de máquinas de Turing, se puede dar lugar a estructuras más complejas y poderosas. Pérez, en su definición de máquinas de Turing, las describe como dispositivos diseñados para el reconocimiento de lenguajes. Estos dispositivos consisten en una colección de celdas de almacenamiento que se extiende infinitamente en ambas direcciones, conformando así una cinta infinita. Esta conceptualización va más allá de la mera identificación de patrones lingüísticos y establece las bases teóricas para comprender la esencia misma de la computación.

Algunas de las características de las máquina de Turing de acuerdo con Pérez son las siguientes:

- **Cada celda almacena un único símbolo:** Cada posición en la cinta contiene un único símbolo, que puede ser leído o modificado por la cabeza de lectura/escritura.
- **La colección no tiene una celda primera ni única:** No hay una celda designada como la primera o única celda en la cinta.
- **Tiene capacidad de almacenamiento ilimitada:** La cinta tiene la capacidad de almacenar un número infinito de celdas.
- **A los contenidos de las celdas se puede acceder en cualquier orden:** La cabeza de

lectura/escritura puede moverse libremente sobre la cinta y acceder a los contenidos de las celdas en cualquier orden.

- **La cinta tiene asociada una cabeza de lectura/escritura:** La máquina de cinta infinita tiene una cabeza de lectura/escritura que puede desplazarse sobre la cinta y, en cada movimiento, leer o escribir un símbolo. Esta característica la diferencia de los autómatas finitos y los autómatas de pila. (Pérez, 2005)

Existe una amplia variedad de implementaciones de la máquina de Turing, desde las Máquinas de Turing multicinta hasta las Máquinas de Turing con cabezales múltiples o las Máquinas de Turing cuánticas. En este proyecto utilizaremos una máquina de Turing multicinta, pues se diseñará una máquina de Turing de dos cintas que pueda realizar una operación.

Las máquinas de Turing multicinta son una extensión de las máquinas de Turing clásicas y permiten realizar ciertos cálculos más fácilmente. En lugar de tener una única cinta y una cabeza de lectura/escritura, estas máquinas incorporan varias cintas con cabezas de lectura/escritura independientes. Esto permite realizar múltiples operaciones simultáneamente y agilizar ciertos procesos computacionales.

La principal ventaja de las máquinas de Turing multicinta es su capacidad para realizar cálculos de manera paralela, lo que puede resultar en una mejora de la eficiencia en comparación con las máquinas de Turing estándar aunque la complejidad en el diseño y la implementación de algoritmos para estas máquinas es mayor debido a la gestión de múltiples recursos.

Aunque teóricamente más poderosas, las máquinas de Turing multicinta son equivalentes en capacidad de cómputo a las máquinas de Turing estándar, lo que significa que cualquier problema resoluble por una máquina de Turing multicinta también puede resolverse por una máquina de Turing estándar.

2. Análisis:

Como nuestra máquina de Turing es capaz de aceptar enteros positivos y negativos, dividimos el problema en dos partes:

- Detección de los signos
- Multiplicación

Para resolver el problema, lo primero que hicimos fue abordar cada uno de los problemas de manera individual.



3. Diseño

Nuestra máquina de Turing está definida como una estructura algebraica compuesta por:

$$MT = \{Q, \Sigma, \Gamma, q_0, \delta, F, B\}$$

donde:

- $L = \{n\bar{x}m\}$
- $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$
- $\Sigma = \{1, 0, *, \$, B\}$
- $\Gamma = \{1, 0, *, \$, B\}$
- $q_0 = \{q_0\}$
- $F = \{q_8\}$

Detección de Signos

Como nuestra máquina de Turing es capaz de aceptar enteros positivos y negativos, dividimos el problema en dos partes:

- Positivo por positivo (+)
- Negativo por positivo (-)
- Negativo por negativo (+)

Por lo que imaginamos una máquina de Turing con tres posibles caminos.

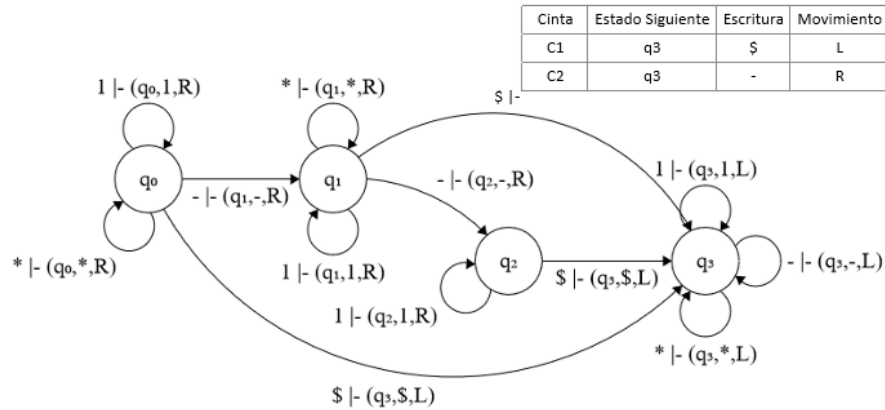


Figura 1: Sección de la MT para detectar signos

Los tres casos antes planteados se recorren en la máquina de Turing de la siguiente manera:

- Positivo por positivo (+)

Recorrerá toda la cinta en q_0 y al detectar el símbolo que usamos para marcar el final se mueve a q_3 .

- Negativo por positivo (-)

Recorrerá parte de la cinta en q_0 y cuando detecte el signo (-) que usamos para marcar los números negativos se trasladará a q_1 y recorrerá el resto de la cinta en q_1 y al no volver a detectar otro (-) en la cinta, detectará el símbolo que usamos para marcar el final y se mueve a q_3 . En este movimiento haremos uso de la segunda cinta y escribiremos en la misma un signo (-) para cuando se complete la multiplicación podamos entender que el resultado es negativo.

- Negativo por negativo (+)

Recorrerá parte de la cinta en q_0 y cuando detecte el signo (-) que usamos para marcar los números negativos se trasladará a q_1 . Cuando se traslade a este estado, seguirá recorriendo la cinta hasta que detecte el segundo (-) y se moverá a q_2 . Al avanzar a este estado, recorreremos el resto de la cinta hasta detectar el símbolo que se usa para marcar el final y se moverá a q_3 .

En todos los casos iniciamos en el estado q_0 y acabamos en el estado q_3 . La única operación destacable es cuando se escribe en la segunda cinta en el segundo caso. Desde aquí, nuestro planteamiento era usar la segunda cinta únicamente para escribir ahí el resultado de la operación y en esta parte de la máquina de Turing solo se escribe en el caso que se detecte en uno de los 2 números el signo (-) que

nos daría como resultado un número negativo.

Multiplicación

Para entender esta máquina de Turing, tenemos que entender cómo procesamos los números antes de meterlos en la primera cinta. Por ejemplo, consideremos la operación 4×2 . Al meterla a la cinta, transformaremos los números decimales en una serie de unos que corresponden al número de la siguiente manera:

1	1	1	1	*	1	1	\$
---	---	---	---	---	---	---	----

Al verlo de esta manera, la máquina de Turing encargada de la multiplicación se ve de la siguiente manera:

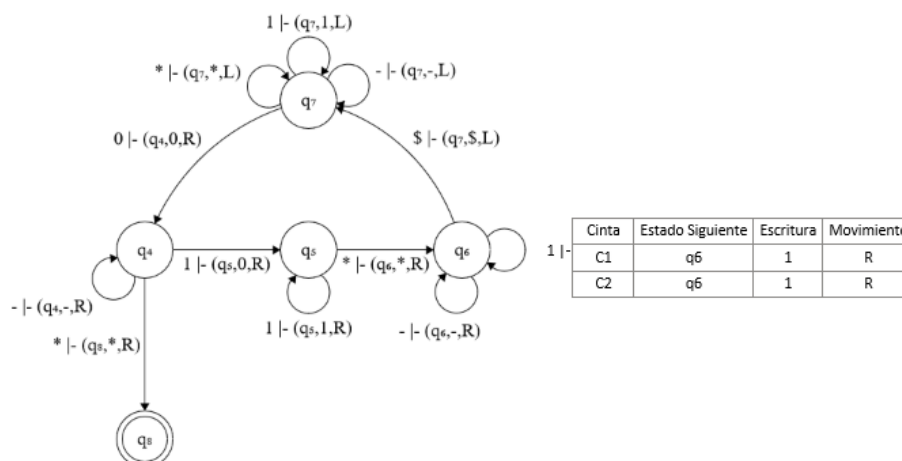


Figura 2: Sección de la MT que realiza la multiplicación

Estando en el estado q_4 , leemos el primer 1 y lo marcamos como cero. Tras eso, avanzamos al estado q_5 y recorremos la cinta hasta encontrar el signo (*) y nos pasamos al estado q_6 . Dentro de este estado, la transición importante es la que detecta los 1, ya que por cada 1 escribiremos un 1 en la segunda cinta. Para ejemplificar hasta este punto, se vería de esta manera las cintas.

Cinta 1:

0	1	1	1	*	1	1	\$
---	---	---	---	---	---	---	----

Cinta 2:

1	1
---	---

Tras esto, pasaremos al estado q_7 , que sirve para regresar hasta encontrar un cero, que representa



	Cinta 1						Cinta 2							
	1	*	-	\$	0	B	1	*	-	\$	0	B		
q ₀	q ₀ 1 R	q ₀ * R	q ₁ - R	q ₃ \$ L								q ₁ B R	q ₃ B R	
q ₁	q ₁ 1 R	q ₁ * R	q ₂ - R	q ₃ \$ L								q ₂ B R	q ₃ - R	
q ₂	q ₂ 1 R			q ₃ \$ L								q ₃ B R		
q ₃	q ₃ 1 L	q ₃ * L	q ₃ - L			q ₄ B R						q ₄ B R		
q ₄	q ₅ 0 R	q ₅ * R	q ₄ - R									q ₅ B R	q ₅ B R	
q ₅	q ₅ 1 R	q ₅ * R										q ₅ B L		
q ₆	q ₆ 1 R		q ₆ - R	q ₇ \$ L								q ₆ 1 R	q ₇ B R	
q ₇	q ₇ 1 R	q ₇ * R	q ₇ - R		q ₄ 0 R							q ₄ B L		
q ₈														

Figura 4: Tabla de transición

4. Funcionalidad:

Para comprobar el correcto funcionamiento de nuestra solución con la máquina de Turing de doble cinta, se llevó a cabo una abstracción de nuestro diseño para posteriormente codificarlo en lenguaje C. Para implementar esta abstracción de manera correcta, se hizo uso de diversos conceptos y herramientas de programación, desde el uso de funciones y estructuras hasta la utilización de un Dev Container.

Un Dev Container es un entorno de desarrollo aislado y reproducible que se ejecuta en un contenedor de Docker, con un Dev Container, ya no es necesario instalar el compilador GCC en la máquina local. Para utilizar un devcontainer, se debe tener instalado Docker en la máquina local.

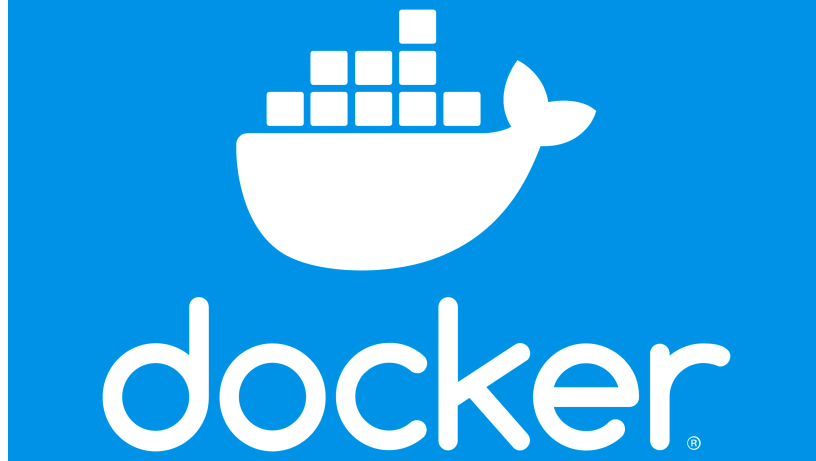


Figura 5: Logo Docker de: <https://logos-marcas.com/docker-logo/>

A continuación, se proporciona el enlace de acceso al repositorio donde se encuentran todos los elementos necesarios para ejecutar el programa:

https://github.com/Eickyair/maquina_turing/tree/main

Se escribió un readme con todas las instrucciones necesarias para ejecutar el programa de la Máquina de Turing de doble cinta que realiza la función de multiplicación, adicionalmente se incluyeron las instrucciones para usar un DevContainer que se configuró para ejecutar correctamente el programa.

Funcionamiento del código

A continuación se mostrará una breve explicación de los componentes más importantes del programa, los cuales se pueden encontrar con una explicación más detallada en la documentación del repositorio de GitHub.

Estructuras:

Autómata

Abstracción fundamental que almacena la información necesaria para representar un autómata de Turing, incluyendo el número de estados finales, el estado inicial, los estados finales, el número de elementos del alfabeto de entrada, el alfabeto de entrada, el número de elementos del alfabeto de la cinta, el alfabeto de la cinta, el número de estados y la función de transición. También almacena dos cintas, que son utilizadas por el autómata para realizar las operaciones.

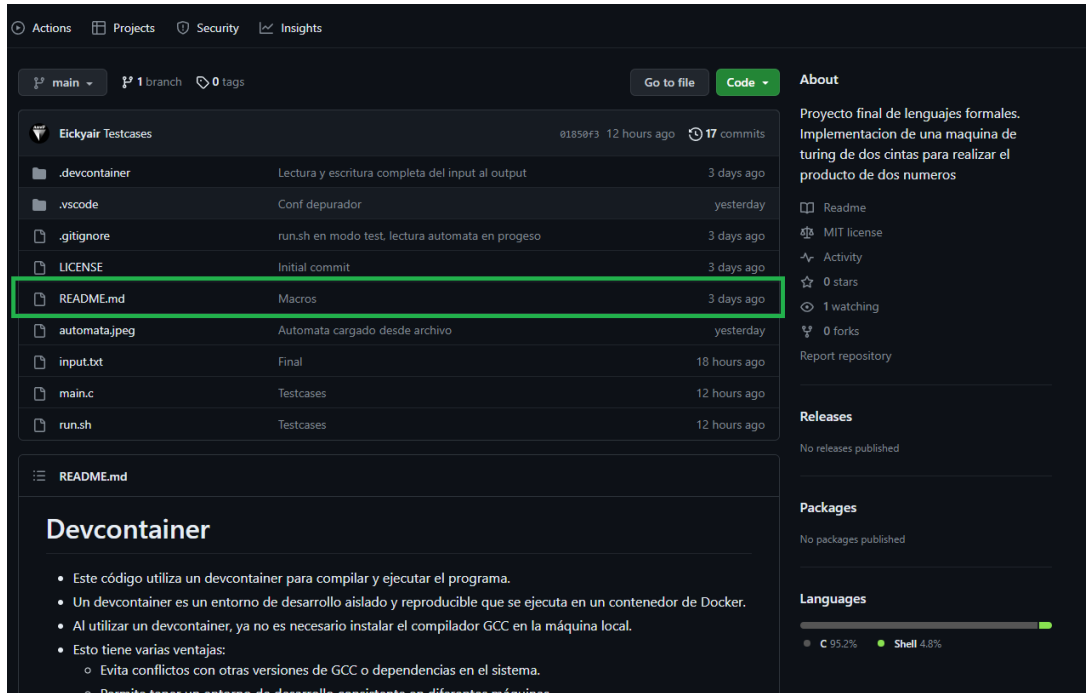


Figura 6: Repositorio en GitHub.

Cinta

Estructura fundamental que representa una cinta de la Máquina de Turing, se utiliza para representar el estado actual de la cinta en una Máquina de Turing y facilita la manipulación de los símbolos y la posición del cabezal.

Arista

Estructura fundamental que nos ayuda a representar una arista para un nodo de la máquina de Turing, contiene el símbolo que se lee, el estado destino y el número de operaciones de escritura en la cinta.

Estado

Es una estructura que representa un estado actual de la máquina de Turing de doble cinta.

OperacionEscrituraCinta

Es una abstracción que representa la escritura en alguna de las cintas de nuestra Máquina de Turing.

Ejecución

Para ejecutar el programa, se hace uso del archivo run.sh, en el cual se encuentra un script con las instrucciones necesarias para correr el programa. Se realizaron los cambios necesarios para que la entrada del programa no sea un input del teclado, sino un archivo llamado input.txt. De igual forma,

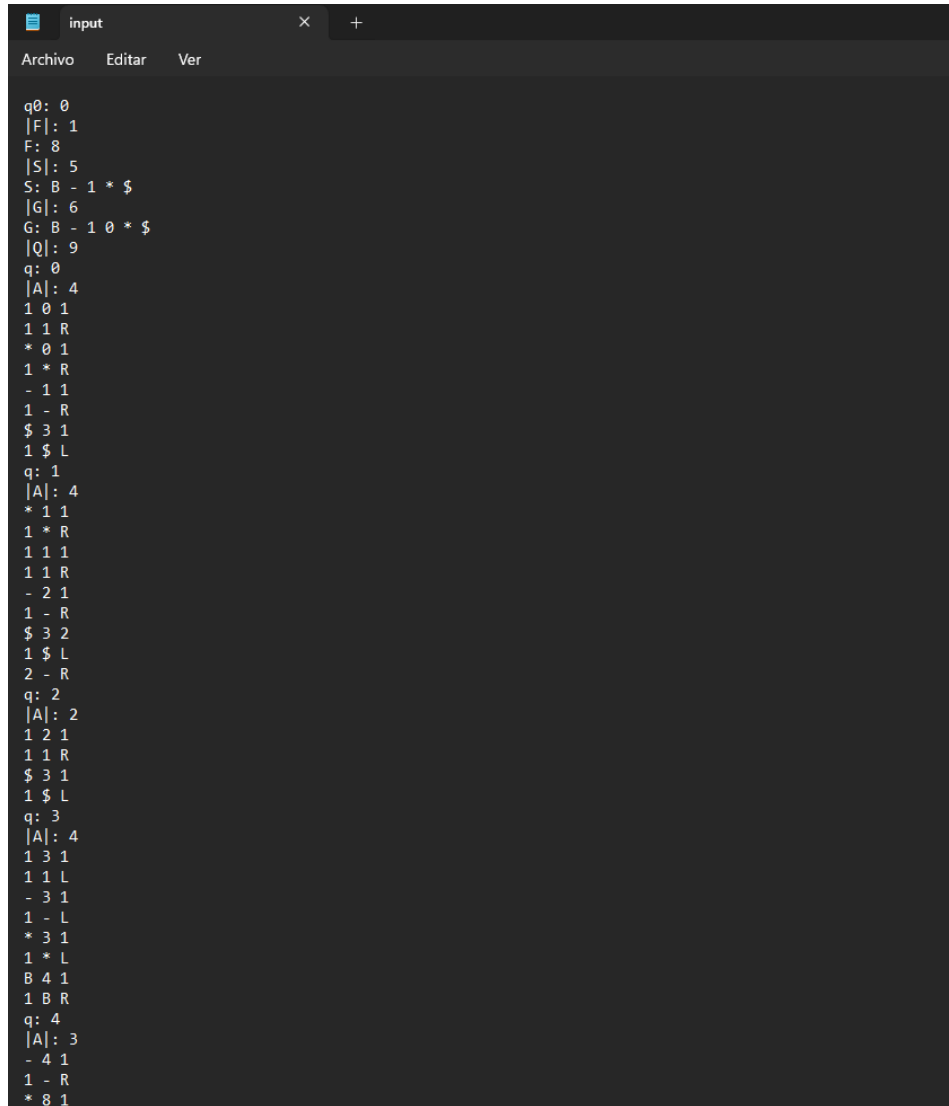
se hicieron los cambios para que la salida no se despliegue en la terminal, sino en un archivo llamado output.txt.

```
# Verifica si la compilación fue exitosa
if [ $? -eq 0 ]; then
    # Verifica si se recibió la bandera -t
    if [ "$1" = "-t" ]; then
        # Ejecuta el programa "main" con el archivo de entrada "test.txt" y redirige
        # Asi no se afecta el input.txt y test.txt es un archivo local para cada desarrollador
        # la salida al archivo "output.txt".
        ./main < test.txt > output.txt
    else
        # Ejecuta el programa "main" con el archivo de entrada "input.txt" y redirige
        # la salida al archivo "output.txt".
        ./main < input.txt > output.txt
    fi
fi

# Verifica si la ejecución del programa fue exitosa
if [ $? -eq 0 ]; then
    # Elimina el archivo ejecutable "main"
    rm ./main
```

Figura 7: rush.sh

A continuación podemos ver el archivo input.txt. En la primera parte se introducen las entradas del automata que se definió al inicio, para realizar las operaciones de una Máquina de Turing de doble cinta para una multiplicación.



```
q0: 0
|F|: 1
F: 8
|S|: 5
S: B - 1 * $
|G|: 6
G: B - 1 0 * $
|Q|: 9
q: 0
|A|: 4
1 0 1
1 1 R
* 0 1
1 * R
- 1 1
1 - R
$ 3 1
1 $ L
q: 1
|A|: 4
* 1 1
1 * R
1 1 1
1 1 R
- 2 1
1 - R
$ 3 2
1 $ L
2 - R
q: 2
|A|: 2
1 2 1
1 1 R
$ 3 1
1 $ L
q: 3
|A|: 4
1 3 1
1 1 L
- 3 1
1 - L
* 3 1
1 * L
B 4 1
1 B R
q: 4
|A|: 3
- 4 1
1 - R
* 8 1
```

Figura 8: Input.txt



Al final del archivo, se tienen 4 pruebas diferentes para la Máquina de Turing que se definio previamente. La primera prueba genera un input de los números 3 y 4, la segunda de los números 3 y -4, la tercera de los numeros 0 y -0, finalmente la cuarta recibe los números -0 y 0.

```
Archivo Editar Ver
1 1 1
1 3 1
1 1 L
- 3 1
1 - L
* 3 1
1 * L
B 4 1
1 B R
q: 4
|A|: 3
- 4 1
1 - R
* 8 1
1 * R
1 5 1
1 0 R
q: 5
|A|: 2
1 5 1
1 1 R
* 6 1
1 * R
q: 6
|A|: 3
- 6 1
1 - R
$ 7 1
1 $ L
1 6 2
1 1 R
2 1 R
q: 7
|A|: 4
1 7 1
1 1 L
- 7 1
1 - L
* 7 1
1 * L
0 4 1
1 0 R
q: 8
|A|: 0
t: 7
-2 -22c1 0
AX C2 0
z z 0
3 4 1
3 -4 1
0 -0 1
-0 0 1
```

Figura 9: Input.txt

Finalmente, se ejecutó el programa con las pruebas correspondientes para la Máquina de Turing de doble cinta y se corroboró su correcto funcionamiento.



```
TEST_CASE :: 1: OK
TEST_CASE :: 2: OK
TEST_CASE :: 3: OK
TEST_CASE :: 4: OK
```

Figura 10: Output.txt

5. Conclusiones

Aguilar Martínez Erick Yair:

El proyecto me pareció un reto interesante; sin embargo, hubo bastantes desafíos. Uno de ellos fue determinar la teoría relacionada con las máquinas de Turing de varias cintas, ya que no hay mucha información al respecto. No obstante, a medida que se investigaba el tema, estas dudas se fueron resolviendo junto con los antecedentes que la materia nos proporcionó.

El otro problema estuvo relacionado con el lenguaje de programación, que en ocasiones presentaba errores en cuanto a la reserva de memoria. Estos errores debían depurarse paso a paso para determinar en qué sección se encontraba el error.

Al final del curso, veo que los programas se pueden representar como una serie de estados y transiciones, cada uno con sus respectivos flujos. Este enfoque me resulta mucho más sencillo y descriptivo, y le da más sentido a por qué muchas veces representamos los algoritmos con diagramas de flujo, ya que es así como opera la computadora, con transiciones y estados.

Con respecto a mi equipo de trabajo, es muy cómodo trabajar con ellos, y sin duda, me gustaría encontrarme con ellos en el futuro.

López Ramírez Monserrat:

Con base en los resultados obtenidos, podemos concluir que las Máquinas de Turing de doble cinta permiten el reconocimiento de lenguajes. En este caso, nos permitieron reconocer la función de una multiplicación al introducir los tokens correspondientes, los cuales estuvieron compuestos por números enteros signados.

El análisis y el diseño del autómata fueron los pasos más importantes de este proyecto, puesto que a partir de un correcto diseño del mismo se tienen más posibilidades de continuar con los demás pasos



de manera exitosa. Por otro lado, si desde el inicio el autómata no se plantea correctamente, todo lo demás estará diseñado e implementado con errores.

La implementación de una máquina de Turing en lenguaje C permitió corroborar el funcionamiento de nuestra Máquina de Turing con diferentes entradas. Además, se reconoció la importancia del uso de la programación como herramienta para implementar y automatizar el uso de diferentes algoritmos.

Ceniceros Mariaca Carlos:

En líneas generales, me pareció un buen proyecto para finalizar el curso ya que me ayudó a implementar lo visto en el curso, sobre todo los temas vistos en la parte final del curso. Lo más desafiante de este proyecto fue el ver cómo implementar una operación más compleja dentro de una máquina de Turing, ya que a lo largo de las clases manejamos expresiones más sencillas que nos sirvieron para comprender el funcionamiento de esta estructura.

El proyecto, al estar dividido en tres partes, fue más fácil de digerir, pero el avanzar de una fase a otra fue un desafío en sí, sobre todo en implementar la máquina de Turing al lenguaje de programación.

Referencias

- [1] Pérez, I. (2005). *Lenguaje y Compiladores*. Universidad Católica Andrés.

Los créditos de las fotografías pertenecen a sus respectivos autores.