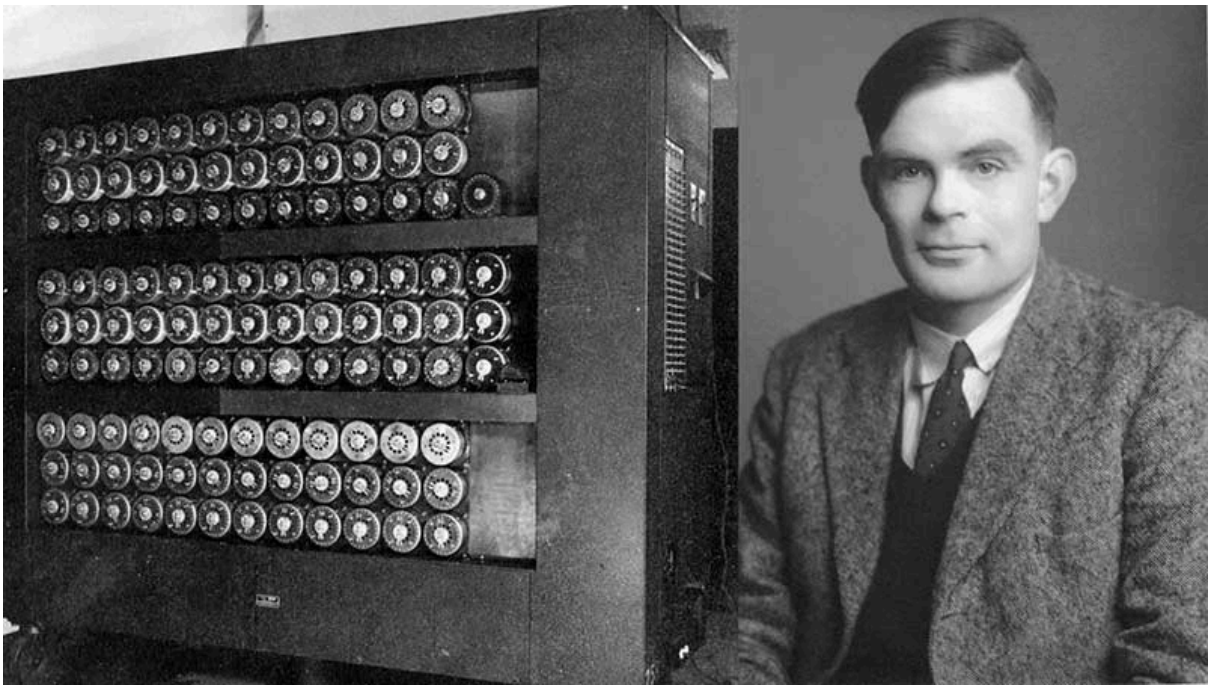
TAREA 1 (24-25)



DISEÑO DE MÁQUINAS DE TURING

Adrián Yared Armas de la Nuez

Índice

Introducción

Las máquinas de Turing, propuestas por el matemático y lógico británico Alan Turing en 1936, son un modelo abstracto de computación que sirve para comprender los fundamentos teóricos de los algoritmos y la computabilidad. Consisten en una cinta infinita que actúa como memoria, un cabezal que puede leer y escribir símbolos, y un conjunto de reglas que determinan las acciones del cabezal según el símbolo leído. Este concepto ha sido fundamental para el desarrollo de la informática moderna, ya que establece los límites de lo que puede ser computado. Además, las máquinas de Turing son una herramienta clave en la teoría de la complejidad computacional.

1.Enunciado

Crea una máquina de Turing capaz de voltear una secuencia de entrada de n bits tal que n>=1.
Por ejemplo 111001101011 110101100111

1.1.Mi desarrollo en el problema (Introducción)

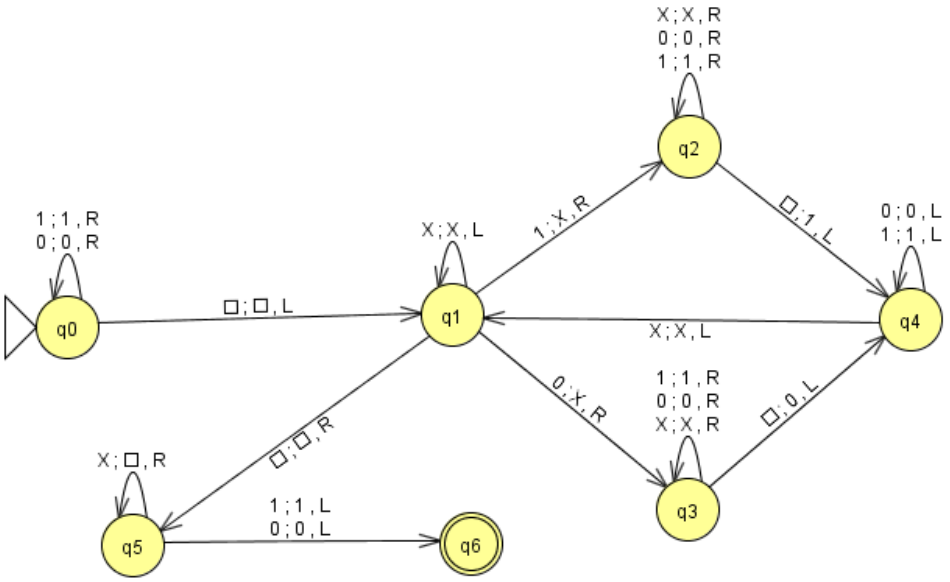
Durante esta actividad he pasado por tres fases, dos previas al resultado final debido a la “limitación” de uso de las máquinas de Turing a primera vista debido a su sencillez de instrucción y mi falta de experiencia.

1.1.1.Volteo y primer problema

En esta primera fase logré voltear la string pero me topé con el principal problema de esta actividad, la falta de llegada a un estado final o bucle.

1.1.2.manera con variables de apoyo (no lenguaje Turing)

Tras plantearme cómo llegar a romper ese bucle decidí hacerlo de manera progresiva, enfrentándose primero a una versión más sencilla del problema a través del uso de variables de apoyo, esta manera no está correcta ya que usas unos caracteres que no pertenecen al lenguaje de la máquina de Turín, la cual cuenta con 0,1 y Blank.



Como se puede apreciar en el diagrama, en esta máquina se recorren todos los 0 y 1 hasta llegar al blank, posteriormente si se encuentra un 0 o un 1, este se sustituye por la variable auxiliar X y se avanza un paso a la derecha donde se copia el número anteriormente sustituido. Posteriormente repetimos en bucle lo siguiente, se va a la izquierda ignorando tantas x como haya hasta llegar a un 0 o 1 y se repite la operación anterior (se pone una X, se recorre la cadena hasta el final de la derecha y se pone en Blank el número). Este bucle termina cuando al recorrer la cadena hacia la izquierda en vez de encontrar un 0 o 1 encontramos un blank. En ese caso recorreremos la cadena de nuevo a la derecha hasta llegar al primer 0 o 1, pero esta vez sustituyendo las X por blank. Una vez llegues al primer 0 o 1 llegarás al estado final.

1.2.1.Resultado final

Para esta versión final he utilizado solo el lenguaje de Turing y me he planteado un pequeño resumen previo al inicio del desarrollo en jflap, lo dejo a continuación:

1. Ve al final de la cadena de 0 y 1 introducida por teclado. A esta posición le llamaré N. cadena de ejemplo (10111)

2. Ve a n-1, si encuentro un 1 ve dos a la derecha y escribe 1, y si es 0 ve dos a la derecha y escribe 0.
Ahora tendría algo tal que así 1011B1

3.Después debo ir dos a la izquierda, si es 1 pon un blank y ve a N y recorre tantos 1 y 0 como haya hasta llegar a blank y pon el 1, y si es un 0, realiza lo mismo pero con un 0.

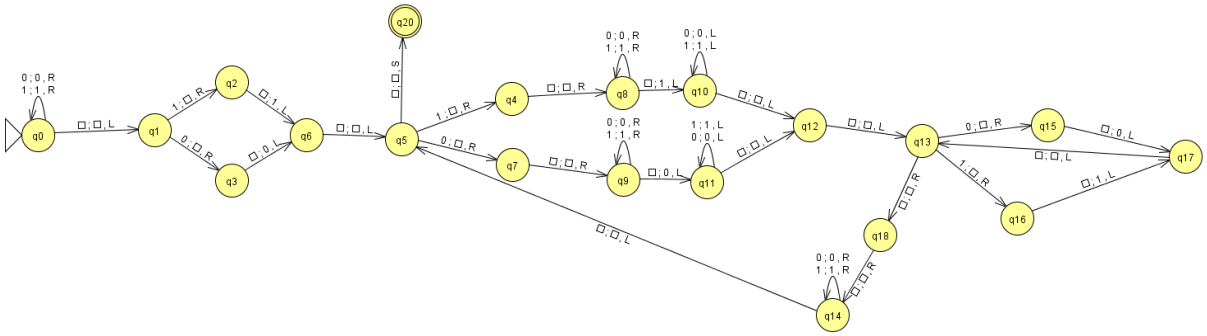
4. A continuación debo ir a n y realizar en bucle la siguiente acción: va dos a la izquierda y si es 1 pon el hueco en blanco, va uno a la derecha y pon 1. Si es 0 has lo mismo pero con 0. Y en el caso de encontrar un Blank deja de hacer el bucle y va hacia la derecha hasta encontrar un N.
Con esto tendré algo tal que 101B11

5. Ahora debo realizar de nuevo los pasos 2, 3 y 4, de manera que quedaría de la siguiente manera:
10BN111 -> 1BN111 -> 1N1110
1N1110 -> BN11101 ->N11101
11101

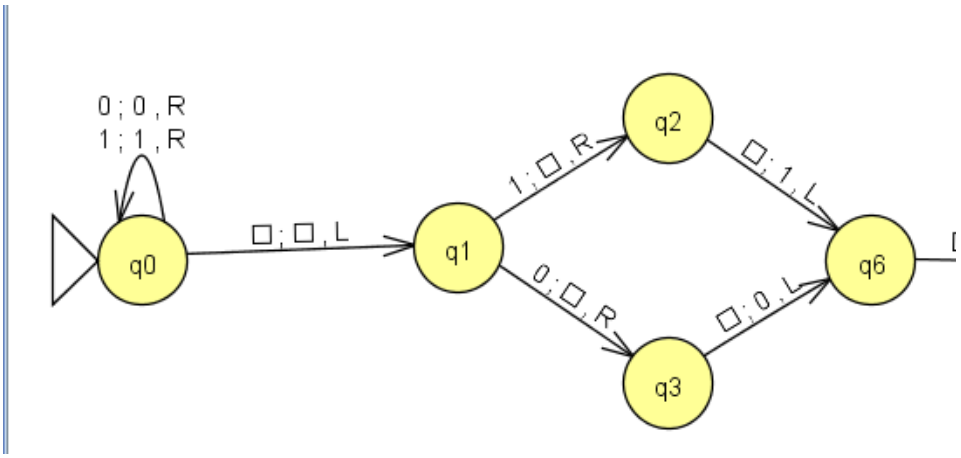
7. Una vez volteado, debo llegar al estado final ya que al recorrer la cadena hacia la izquierda se puede cumplir una condición que antes era imposible, encontrar tres Blanks seguidos. BBBN11101

1.2.2. Diagrama y explicación

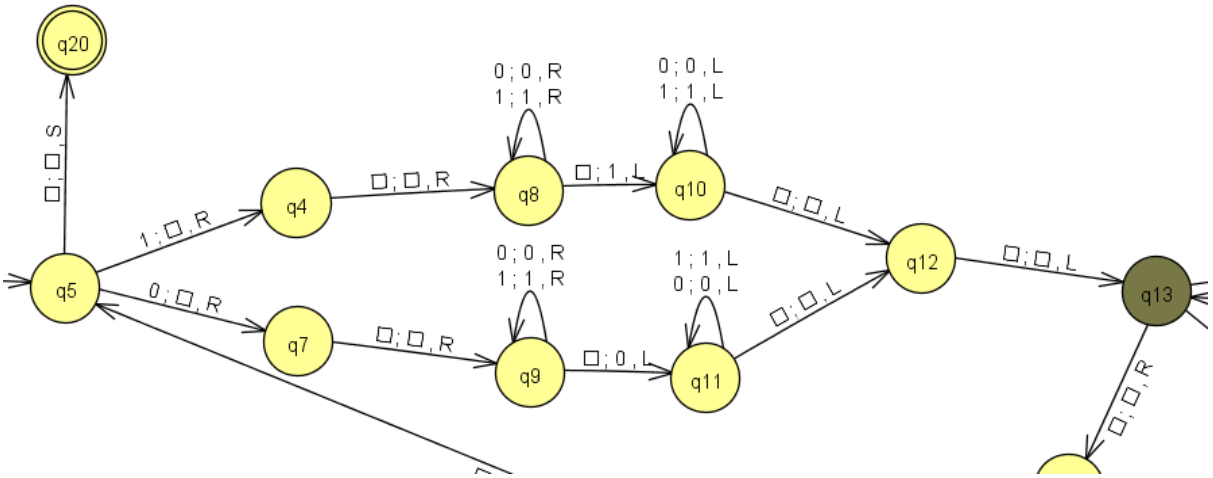
Para la versión final, con una noción más clara del problema, he realizado lo siguiente:



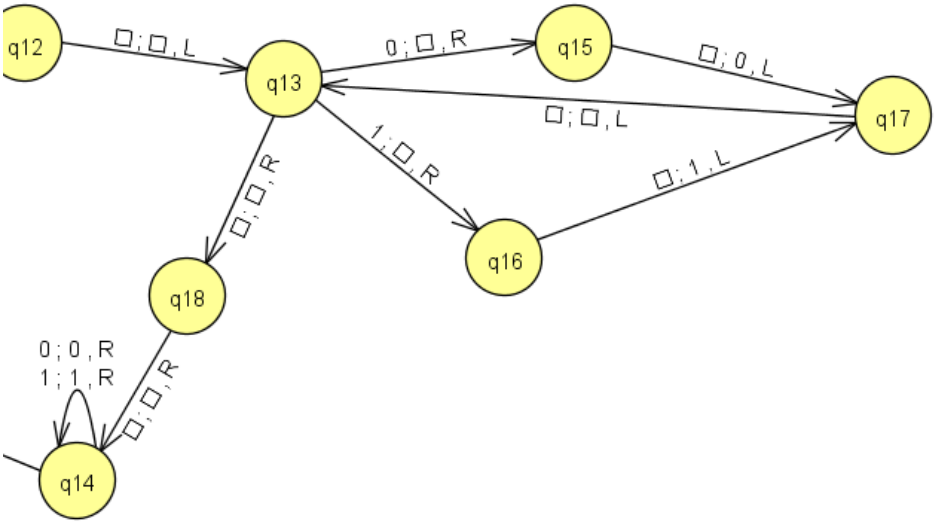
Iré explicando a continuación cada parte:



N es el nombre que le he dado a la primera posición Blank de la derecha de la cadena de 0 y 1.
q0 recorre la cadena hasta encontrar un blank, posteriormente al llegar a q1 si se encuentra un 1 o un 0 se sustituye en N-1 por blank y en N+1 por el 0 (q3) o 1(q2) que había anteriormente. A este punto tendríamos algo tal que así 1110N1 y estaríamos en q6.



Después iríamos a q5 y un hueco a la izquierda 111**0**N1, e iríamos a q7 o a q4 dependiendo de si encontramos un 1 o un 0, sustituiremos el valor por Blank e iremos a N+1 y en q9 o q8 iremos hasta el final de la cadena invertida y pondremos el valor que habíamos sustituido anteriormente. (111N10)
Y con q10 o q11 y con q12, iremos a N-2.



A continuación, dependiendo de si encontramos un 1 o un 0 iremos a q15 o q16 y sustituiremos en número por un blank y del paso de q15 y q16 a q17 pondremos el valor en la casilla e iremos un paso a la izquierda y al encontrar el blank iremos a q13 de nuevo.
Una vez aquí si encuentra un 1 o 0 repetirá la operación y si no volverá al inicion de la cadena original con q18 y a N con q14

Ahora nos encontramos en N-1 al pasar a q5 y volveremos a repetir;

el paso de copia por q4 ->q8 -> q10 -> q12 o por q7 -> q9 -> q11 -> q12,: y el paso de ordenado por el bucle de q13, q14, q16 y q17.
Todo esto hasta acabar sin números en las posiciones anteriores a N.

Una vez estemos en q5 y cumplamos con esa condición, iremos a q20 debido a la falta de 0 y 1 a nuestra izquierda.

1.3.Pasos de la máquina visualmente

11101N -> 1110N1-> 111BN10 -> 11B1N10 -> 1B11N10 -> 111N10
11BN101 -> 1B1N101 -> 11N101 -> 1BN1011 -> 1N1011 -> N10111

1.4.Ejemplo de ejecución múltiple:

Table Text Size

Input	Result
1101	Accept
0111	Accept
0101011010111	Accept
01101000000000001	Accept
01010101010100010101	Accept
010101011111101001	Accept
10010101111	Accept
1	Accept
0	Accept

2. Infórmate sobre Casos de Uso de la Máquina de Turing

Las máquinas de Turing, propuestas por Alan Turing, son un modelo teórico fundamental en la computación que permite entender y formalizar la noción de algoritmos y computabilidad. A lo largo de los años, han encontrado aplicaciones en diversos campos, desde la teoría de la complejidad y la criptografía hasta la inteligencia artificial y la educación en computación. Estos casos de uso no solo ilustran su importancia teórica, sino también su relevancia práctica en el desarrollo de tecnologías y soluciones informáticas. A continuación, se presentan algunos de los principales casos de uso de las máquinas de Turing:

Teoría de la Computación: Definen formalmente qué es un algoritmo y determinan los límites de lo que se puede computar.

Lenguajes Formales y Autómatas: Sirven para clasificar lenguajes de programación y demostrar su Turing-completitud (aquel que puede hacer la máquina de Turing).

Inteligencia Artificial y Aprendizaje Automático: Se utilizan para simular algoritmos y analizar estrategias de toma de decisiones en contextos complejos.

Teoría de la Complejidad: Ayudan a clasificar problemas en diferentes clases de complejidad, como P y NP, y a entender su dificultad.

Criptografía: Se aplican para evaluar la seguridad de algoritmos criptográficos basándose en problemas difíciles de resolver.

Educación en Computación: Se utilizan como herramienta didáctica para enseñar conceptos fundamentales de computación y algoritmos.

Investigación en Matemáticas: Exploran funciones recursivas y fundamentan muchas pruebas sobre la computabilidad.

Desarrollo de Software: Se aplican en la verificación formal de programas para garantizar propiedades específicas y su correcto funcionamiento.