

TeoriaComputacion_8

February 19, 2020

Teoría de la Computación

Máquinas de Turing

Prof. Wladimir Rodriguez

wladimir.rodriguez@outlook.com

Departamento de Computación

0.1 Introducción

0.1.1 Alan Turing (1912 - 1954)

- Una pregunta acechaba a Turing, y era el hecho de que: ¿Debe existir al menos en principio algún método definido, o proceso mediante el cual toda cuestión matemática pueda ser demostrada? (entscheidungsproblem)
- Para contestar a esta pregunta necesitaba una definición del concepto método, y para ello analizó que era lo que hacía una persona para transformar un proceso metódico, y buscar una forma de hacer esto mecánicamente. Expresó el análisis en términos de una máquina teórica que sería capaz de transformar con precisión operaciones elementales previamente definidas en símbolos en una cinta de papel. En Agosto de 1936 presenta el concepto final de la Máquina de Turing en su artículo “*On computable numbers, with an application to the Entscheidungsproblems*” (1936).
- En este artículo determinó la naturaleza y las limitaciones teóricas de las máquinas lógicas antes de que se construyera siquiera una sencilla computadora por completo programable
- En Princeton, desarrolló una máquina de cifrado, y estudió sobre este campo debido a la utilidad de ello en la Guerra con Alemania.
- Trabajando secretamente para el Colegio de Cifrado y Código Gubernamental o también llamado Departamento de Criptoanálisis. Turing fue reclutado por Inglaterra, en Bletchley Park, para descifrar los mensajes que componía la máquina alemana Enigma, y, como consecuencia, los aliados construyeron la máquina Colossus.
- Es en este periodo cuando toma contacto con la más avanzada tecnología electrónica de la época y planea la Máquina de Turing Universal en su forma electrónica, de hecho había inventado las computadoras digitales.

- En 1950, Turing publica el artículo *Computing Machinery and Intelligence* en la revista Mind, en el que introducía el célebre Test de Turing. Este artículo estimuló a los pensadores sobre la filosofía e investigación en el campo de la Inteligencia Artificial.
- Turing tuvo la visión de una computadora con memoria que implementaría las funciones aritméticas mediante programación en vez de con componentes electrónicos, y que podría desempeñar todo tipo de tareas (por ejemplo, manejo de archivos, álgebra, jugar ajedrez, encriptamiento, etc.) Hacia 1947, concibió la idea de las redes de cómputo y el concepto de subrutina y biblioteca de software.
- En vez de publicar los principios fundamentales de cómputo que había descubierto, se dedicó a estudiar fisiología y neurología, y en un reporte interno del NLP (Laboratorio Nacional de Física, Inglaterra) de esa época, describe las ideas básicas de lo que hoy se conoce como una red neuronal.

0.1.2 ¿Qué es computabilidad?

- Consiste en ser capaz de encontrar la representación adecuada para la descripción de un problema o fenómeno.
- Para tal representación es necesario:
 - Un conjunto finito de símbolos.
 - Hacer asociaciones entre conceptos y elementos del lenguaje (de símbolos)
 - Encontrar las combinaciones adecuadas de símbolos para evitar ambigüedad.
 - Definir una manera de confirmar tal descripción para que terceros puedan reproducirla y llegar a los mismos resultados.

0.2 La máquina de Turing

- En el artículo On Computable Numbers, Turing construyó un modelo formal de computador, la Máquina de Turing, (con esto resolvió el entscheidungsproblem (planteado por, David Hilbert) y demostró que había problemas tales que una máquina no podía resolver. La máquina de Turing es el primer modelo teórico de lo que luego sería un computador programable. Con el tiempo a este tipo de máquina se la conoció como máquina de estado finito, debido a que en cada etapa de un cálculo, la siguiente acción de la máquina se contrastaba con una lista finita de instrucciones de estado posibles.

0.3 Definición

- La Máquina de Turing consiste en una colección de celdas de almacenamiento que se extiende infinitamente en ambas direcciones, almacenando un símbolo en cada celda. Dicha colección no tiene una celda primera ni última, por lo que posee una capacidad de almacenamiento ilimitada.
- Una máquina de Turing (MT) tiene tres componentes:
 - Una cinta infinita dividida en celdas. Cada celda contiene un símbolo.

- Un cabezal que accede a una celda a la vez, y que puede leer y escribir en la cinta, y puede moverse tanto hacia la izquierda como hacia la derecha.
- El control es casi el mismo que en los Autómatas Finitos: el conjunto de estados y transiciones entre estados.

0.4 Definición Formal

- Una Máquina de Turing se define como una septupla $(Q, A, C, \delta, q_0, \beta, F)$ en la que
 - Q es un conjunto finito de estados
 - A es el alfabeto de entrada
 - C es el alfabeto de la cinta
 - δ es la función de transición

$$\delta : Q \times C \rightarrow Q \times C \times (R, L)$$

- q_0 es el estado inicial
- β es el símbolo blanco ($\beta \in C$ y $\beta \notin A$)
- F es el conjunto de estados finales

0.4.1 Video de una implementación de la Máquina de Turing

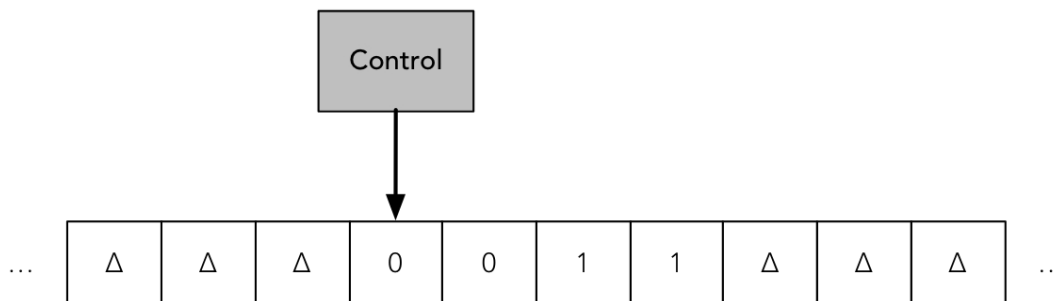
```
[7]: from IPython.lib.display import YouTubeVideo
      YouTubeVideo('E3keLeMwfHY')
```

[7]:



0.4.2 La Cinta

- Asumimos una cinta infinita bidireccional que se extiende hasta el infinito en ambas direcciones.
- Δ denota una celda vacía o en blanco.
- La entrada comienza en la cinta rodeada por Δ con el cabezal en el símbolo del extremo izquierdo. (Si la entrada es ϵ , la cinta está vacía y el cabezal apunta a la celda vacía).



0.4.3 El programa

- El programa de una *Máquina de Turing* es una función de transición; Dependiendo del símbolo debajo del cabezal y el estado, la *Máquina de Turing*:
 - escribe un símbolo
 - se mueve hacia la izquierda o hacia la derecha o permanece en su lugar, y
 - actualiza su estado
- El lenguaje de una *Máquina de Turing* es el conjunto de cadenas que acepta.
- Al igual que los *Autómatas a Pila*, una vez que una *Máquina de Turing* entra en un estado de aceptación, se detiene y termina de manera anormal si no hay transición.

0.4.4 El Diagrama

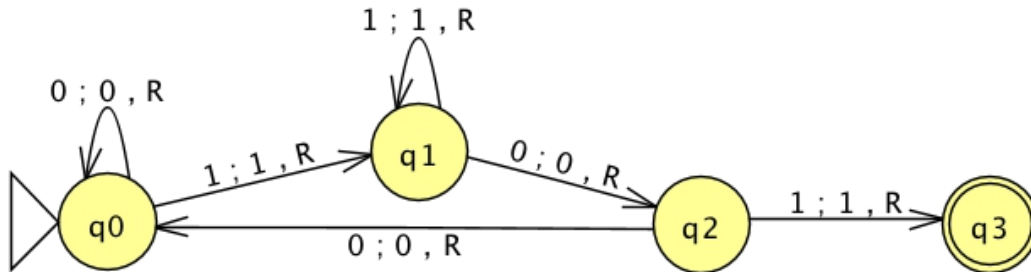
- La *Máquina de Turing* se representa como un diagrama, como para *Autómata Finito*, excepto que cada flecha está etiquetada con un tripleta:

$simboloActual ; simboloNuevo , dirMover$

- donde `dirMover` es uno de *I* (mover hacia la izquierda una celda), *D* (mover hacia la derecha una celda) y *S* (permanecer en su lugar).
- Por ejemplo, triple `01I` significa `si lee un 0, escriba un 1 y mueva el cabezal hacia la izquierda`.

0.4.5 Ejemplo 1: Reconocer la subcadena 101

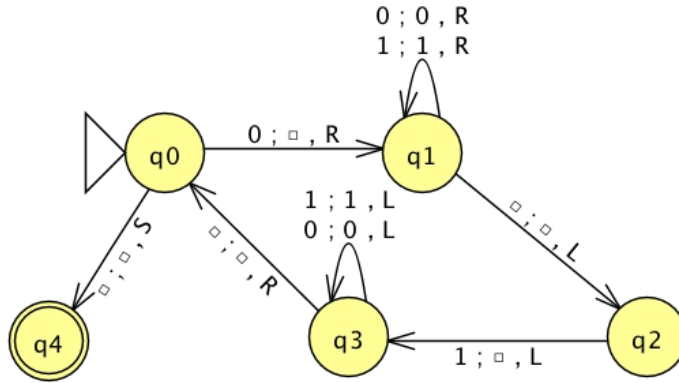
- Aquí hay una *Máquina de Turing* simple que imita un *Autómata Finito* para el lenguaje de todas las cadenas binarias que contienen la subcadena 101.



0.4.6 Ejemplo 2: Reconocer el Lenguaje $0^n 1^n$

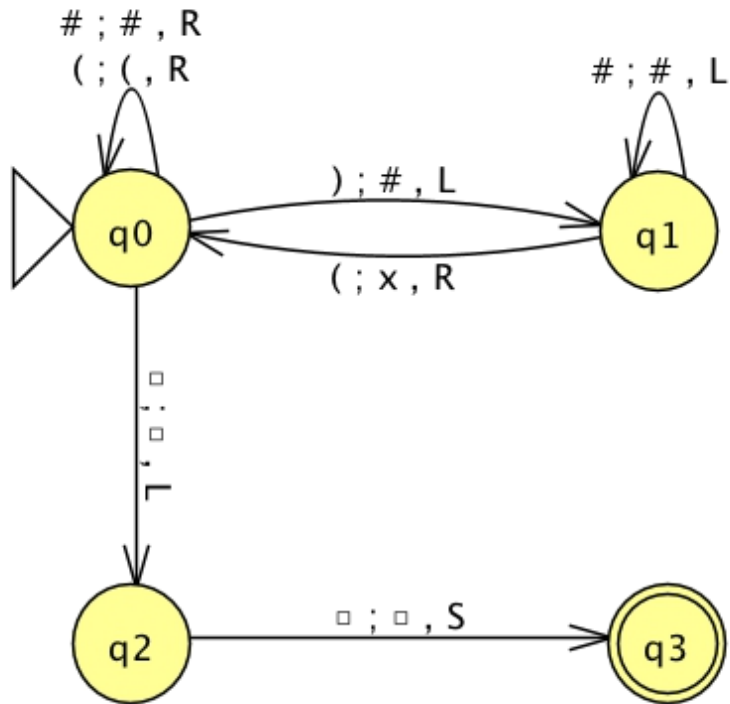
- Para una *Máquina de Turing* que acepta $\{0^n 1^n\}$, empareje los 0 y 1, borre repetidamente el primer 0 y el último 1 hasta alcanzar ϵ . En pseudocódigo:
 1. Si SimboloCabezal = 0, entonces Escribir de lo contrario Rechazar.
 2. Mueva el cabezal hacia la derecha hasta SimboloCabezal = Δ .
 3. Mueva el cabezal hacia la izquierda.
 4. Si SimboloCabezal = 1, entonces Escribir Δ de lo contrario Rechazar.
 5. Mueva el cabezal hacia la izquierda hasta SimboloCabezal = Δ .
 6. Mueva el cabeza hacia la derecha.
 7. Si SimboloCabezal = Δ , entonces Aceptar.
 8. Ir a 1.

0.4.7 Diagrama



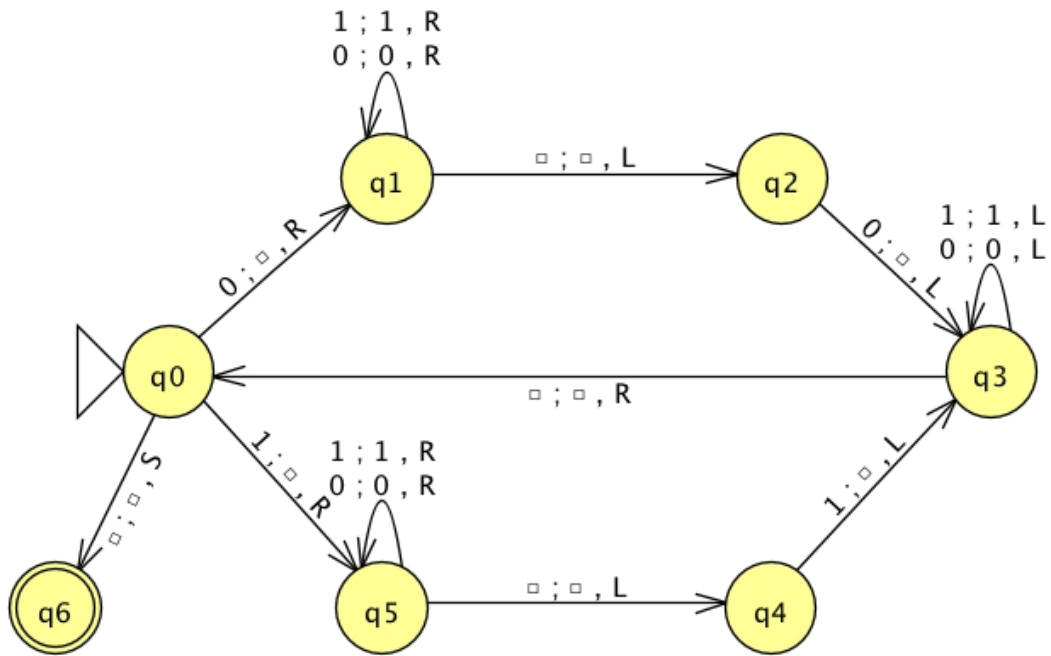
0.4.8 Ejemplo 3: Reconocer Paréntesis Balanceados

- Una *Máquina de Turing* para paréntesis balanceados, una idea es encontrar el par de paréntesis más interno, borrarlos y repetir el proceso. Usamos # para indicar un paréntesis borrado.



0.4.9 Ejemplo 4: Palíndromes

Para palíndromos de longitud par, combinamos el primer y el último símbolo y borramos; luego repetimos. Si se alcanza ϵ sin discordancia, entonces la cadena era un palíndromo.



0.5 Máquinas de Turing como aceptadores de lenguajes

- Sea $M = (Q, A, C, \delta, q_0, \beta, F)$ una Máquina de Turing.
- El lenguaje aceptado por M se define como:

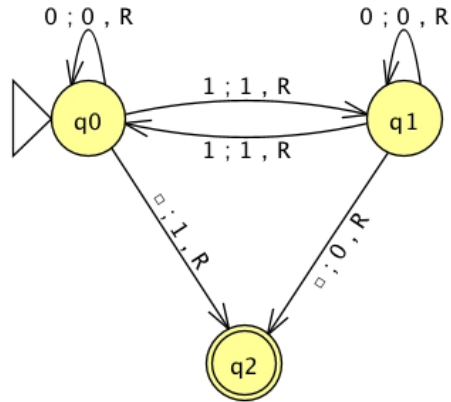
$$L(M) = \{w \in A^* \mid (q_0, az(=w)) \rightarrow^* (p, xby), \text{ para } p \in F, x, y, z \in C^*\}$$

- Diremos, por tanto, que w es aceptada por la Máquina de Turing M , si después de una secuencia de movimientos, la máquina de Turing llega a un estado final y para.

0.6 Cálculos con una Máquina de Turing

- Las máquinas de Turing también se pueden usar para calcular valores
 - Como cálculos aritméticos
 - Por ejemplo, suma, resta, multiplicación, etc.

0.6.1 Ejemplo 5: Calcular Bit de Paridad

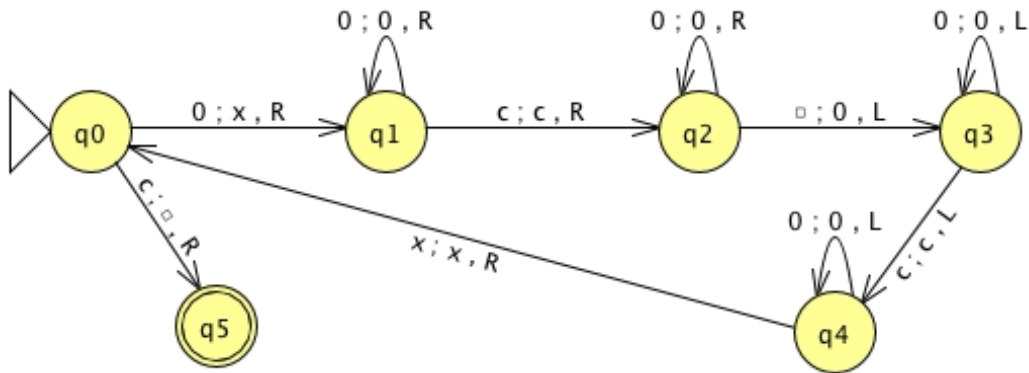


0.6.2 Ejemplo 6: Sumar números

- Un número se representa en formato binario, por ejemplo 5 se representa como (101), pero en caso de las máquinas de Turing se utiliza un formato unario. En formato unario, un número está representado por todos unos o todos ceros. Por ejemplo, 5 estará representado por una secuencia de cinco ceros o cinco unos. $5 = 11111$ o $\$ 0 0 0 0 0 \$$. Vamos a usar ceros para la representación.
- Para sumar 2 números usando una máquina de Turing, estos dos números se dan como entrada a la máquina de Turing separados por una ``c''.

Entrada : 0 0 c 0 0 0 // 2 + 3
Salida : 0 0 0 0 0 // 5

Entrada : 0 0 0 0 c 0 0 0 // 4 + 3
Salida : 0 0 0 0 0 0 0 // 7



0.7 Funciones computables de Turing

- Una función f que convierte cadenas en cadenas es Turing-computable si alguna máquina de Turing M la calcula.
- Es decir, M siempre se detiene, y para la entrada w , M se detiene con $f(w)$ en su cinta.

0.8 Variaciones de las Máquinas de Turing

- La definición de una máquina de Turing es robusta: muchas variaciones no alteran su potencia.
- La idea general es: si se agrega capacidad, entonces demuestre que la Máquina de Turing estándar puede simularla; si se elimina la capacidad, demuestre que la Máquina de Turing resultante puede simular una estándar.
- Máquinas de Turing con almacenamiento
- Máquinas de Turing multipista
- Máquinas de Turing multi-cinta
- Máquinas de Turing no deterministas

0.9 Máquina de Turing no Determinista

- El no determinismo significa que la Máquina de Turing puede tener más de una opción de acción. Como de costumbre, una Máquina de Turing no determinista (o NMT) acepta una cadena si alguna elección de acciones conduce al estado de aceptación.
- Teorema. Una Máquina de Turing no determinista tiene el mismo poder que una Máquina de Turing estándar.

0.10 Tesis de Church-Turing

- Se han propuesto varios modelos de computación a lo largo de los años, pero tienen exactamente el mismo poder que una Máquina de Turing como reconocedores:
- La ``tesis'' de Church-Turing es la creencia/afirmación de que el modelo es apropiado y tiene todo el poder de cualquier computadora que podamos construir.
- Tesis de Church-Turing. Hay un ``procedimiento efectivo'' para un problema si y solo si hay una máquina de Turing para el problema.
- Cualquier cálculo mecánico puede ser realizado por una máquina de Turing
- Hay una máquina de Turing correspondiente a cada problema computable
- Podemos modelar cualquier computadora mecánica con una máquina de Turing
- El conjunto de lenguajes que puede decidir una máquina de Turing es idéntico al conjunto de lenguajes que puede decidir cualquier máquina de computación mecánica.
- Si no hay una máquina de Turing que decida el problema P , no hay un algoritmo que resuelva el problema P .

0.11 Máquina de Turing Universal

- Una máquina de Turing universal es una máquina de Turing que toma otra máquina de Turing como entrada. Para esto, uno necesita especificar una codificación de una máquina de Turing. Las máquinas universales de Turing han sido diseñadas con sorprendentemente pocos estados.