

# Tarea Chica 3: Máquinas de Turing

Profesor Denis Parra  
2 de junio de 2020

---

## Indicaciones

- Fecha de Entrega: Lunes 15 de junio a las 23:59.
  - La entrega de la tarea debe realizarse en el repositorio privado de GitHub Classroom asignado para esta evaluación.
  - Cada hora de atraso o fracción descuenta 5 décimas de la nota que obtengas.
  - La tarea es *individual*. La copia será sancionada con una nota 1.1 en la tarea, además de las sanciones disciplinarias correspondientes.
- 

## Objetivo

Los objetivos de esta tarea son los siguientes:

- Entender el concepto de máquina de Turing.
- Entender la definición matemática formal de una máquina de Turing.
- Entender cómo se construye una máquina de Turing.
- Familiarizarse con el lenguaje  $\text{\LaTeX}$ .

# Introducción

La máquina de Turing es uno de los conceptos más importantes de la teoría de computación ya que consigue formalizar de una forma simple el concepto de computabilidad, o qué cosas son o no computables. La conjetura de Church-Turing postula que para todo proceso computable existe una máquina de Turing que es capaz de realizar el mismo proceso.

Consideramos una máquina de Turing como una cinta infinita de celdas y un cabezal con el que puede leer o escribir la celda en la posición actual. En la configuración inicial, la cinta contiene una palabra (con elementos de un alfabeto de entrada determinado) y el cabezal se encuentra leyendo la primera posición de esta palabra. Las celdas que están antes y después de la palabra contienen un elemento en blanco que representamos como B.

La máquina siempre se encuentra en algún estado  $q \in Q$  el que junto con el elemento que está leyendo el cabezal en ese instante va a determinar la acción que realiza la máquina. Estas acciones se determinan con una función de transición propia de la máquina que toma el elemento en la cinta y el estado, y determina el nuevo estado  $q'$  al que va a pasar la máquina, el elemento que se va a escribir en la cinta y el movimiento del cabezal (que puede ser izquierda, derecha o no moverse).

Formalmente, esta máquina de Turing se define como la tupla

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

donde

$Q$	Conjunto de estados
$\Gamma$	Alfabeto de la máquina
$\Sigma \subsetneq \Gamma$	Alfabeto de entrada
$q_0 \in Q$	Estado inicial de la máquina
$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{\leftarrow, \rightarrow, -\}$	Función de transición
$F \subseteq Q$	Conjunto de estados de aceptación

Cuando una máquina  $M$  se encuentra en un estado  $q$  y un símbolo  $a$ , y no existe una transición  $\delta(q, a)$  decimos que la máquina se detiene. Además, decimos que una palabra  $w$  es aceptada por la máquina  $M$  si la máquina se detiene y además el estado  $q$  en que queda detenida es un estado final ( $q \in F$ ). El conjunto de todas las palabras aceptadas por  $M$  se conoce como el lenguaje definido por  $M$ .

## Ejemplo

La siguiente máquina acepta exclusivamente palabras que contengan solo letras a:

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

donde  $Q = \{q_i, q_f\}$ ,  $\Gamma = \{a, b, B\}$ ,  $\Sigma = \{a, b\}$ ,  $q_0 = q_i$ ,  $F = \{q_f\}$  y la función  $\delta$  define dos transiciones:  $\delta(q_i, a) = (q_i, a, \rightarrow)$  y  $\delta(q_i, B) = (q_f, B, -)$

Notar que  $B$  representa la celda vacía y es un símbolo reservado que puede leer la máquina pero no puede ser usado en el alfabeto de entrada. Esta máquina tampoco altera el contenido de la palabra, porque cada vez que ve una  $a$  escribe esa misma letra, y lo mismo cuando ve un espacio en blanco.

En el alfabeto de entrada solo se permiten los símbolos  $a$  y  $b$  de modo que palabras como  $aaab$ ,  $aba$ ,  $b$ ,  $aaa$  o  $baba$  podrían ser entregadas a la máquina. Sin embargo ninguna de las transiciones define un comportamiento para cuando el cabezal lee una letra  $b$ , por lo que si aparece una  $b$  la máquina se detendrá en un estado no final y por lo tanto la palabra será rechazada. Otra opción es que el cabezal vaya avanzando por la palabra leyendo solo letras  $a$  y cuando llegue al final de la palabra y se encuentre con el símbolo  $B$  cambiaría su estado a  $q_f$  que tampoco tiene transiciones válidas por lo que se detendría. En este último caso la palabra sería aceptada porque  $q_f$  es estado final.

## Actividades

En esta tarea construirás una Máquina de Turing M que acepte el siguiente lenguaje:

$$L = \{n_0.n_1.n_2....n_N \mid n_i = n_{i-1} + n_{i-2} \text{ para } 2 \leq i \leq N, n_0 < n_1\}$$

Y tenga como alfabeto de entrada:

$$\Sigma = \{0, 1, .\}$$

Es decir, secuencias de números binarios separados por un punto que asemejan a la [sucesión de Fibonacci](#), pero donde  $n_0$  y  $n_1$  son números arbitrarios.

Para facilitar la construcción de la máquina, puedes usar los siguientes supuestos:

- La secuencia tendrá al menos tres elementos.
- Todos los números de una secuencia serán distintos.
- $n_i \in \{0, 1\}^k$ , donde k es una constante que depende de la palabra. Es decir, cada  $n_i$  de una secuencia tendrá una longitud fija k (pero k puede variar de una secuencia a otra).
- La máquina M puede tener más de una cinta.

Algunos ejemplos de entradas:

01.10.11 → acepta

0001.0100.0101.1001 → acepta

0010.0001.0011 → rechaza porque  $0010 \not\prec 0001$

0100.1010.1011 → rechaza porque  $0100 + 1010 \neq 1011$

## Implementación (3 puntos)

Primero, debes escribir el código de la máquina M con el formato del simulador online <https://turingmachinesimulator.com/>.

El simulador establece en sus términos y condiciones que ejecuta un *script* para minar criptomonedas. Esto significa que mientras lo estés usando podría aumentar el uso de CPU en tu computador y consumir más rápidamente la batería. Si no estás de acuerdo puedes descargar una extensión en tu navegador para bloquear automáticamente ese tipo de *scripts*. En Google Chrome puedes descargar la extensión que se llama [AdGuard](#)

Se espera que escribas el código de la máquina en el simulador y pruebes varios ejemplos para corroborar que funciona bien.

Se sugiere usar el ejemplo de **suma binaria** que aparece en el simulador como base para esta tarea y revisar [esta página](#) para entender como funcionan las operaciones con números binarios. Está permitido usar fuentes adicionales como ayuda, siempre y cuando sean citadas en el informe.

**Usar código obtenido de internet sin comprender y explicar su funcionamiento será considerado copia, incluso si es citado en el informe.**

## Informe (3 puntos)

Utilizando la plantilla .tex que se encuentra en tu repositorio debes escribir un informe en LaTeX que contenga:

1. (1 punto) Una explicación detallada del funcionamiento de la máquina. Debes explicar los pasos del algoritmo desde que recibe el input hasta que se detiene, incluyendo las decisiones que se toman ante ciertos casos.

La idea de esta parte es que se pueda entender el funcionamiento de la máquina sin necesidad de leer el código. No basta con explicar que hace la máquina, sino que *como* lo hace. Se recomienda usar diagramas que complementen la explicación.

2. (1 punto) Una definición formal para  $Q, q_0, \Gamma, F$ .
3. (1 punto) Una explicación detallada de los estados que componen  $Q$ . Para cada estado se debe incluir:
  - Bajo que condiciones la máquina pasa a este estado.
  - A que estados puede pasar la máquina.
  - Que acciones realiza la máquina mientras se encuentra en este estado.

En caso de varios estados tengan funcionamientos similares, basta con definirlos una vez, pero se debe indicar que estados abarca esta definición.

## Bonus (1 punto)

En esta tarea se puede optar a dos bonus, independientes entre si:

- (0.3 puntos) Si la máquina acepta palabras con números binarios de distinto largo. Es decir, eliminando el tercer supuesto.
- (0.7 puntos) Si la máquina funciona en una sola cinta. Es decir, eliminando el cuarto supuesto.

En caso de querer optar a algún (o ambos) bonus, estos deben ser parte del código de la **única** máquina que entregues. El primer bonus será otorgado automáticamente durante la corrección. Para optar al segundo bonus, debes indicar en la primera parte de tu informe que la máquina funciona con una sola cinta. No se revisarán archivos adicionales, ni archivos que cuenten con más de una máquina.

## Entrega

En resumen, debes entregar en tu repositorio:

- Un archivo de texto simple (.txt) con el código de la máquina, siguiendo el formato del simulador. El nombre del archivo debe corresponder a tu número de alumno (ej: 1563535J.txt).
- Un informe en formato PDF con el nombre informe.pdf, además del archivo .tex que lo compila con el nombre informe.tex. En caso de utilizar archivos adicionales para compilar, también debes incluirlos. Se recomienda utilizar el editor de LaTeX online [Overleaf](#) y el identificador de símbolos [Detexify](#).

## Descuentos

Se aplicarán los siguientes descuentos:

- Hasta 0.5 puntos por informes desordenados o mal escritos.
- 0.5 puntos por mantener los ejemplos de la plantilla en su informe.
- 1 punto por no incluir el archivo .tex
- 0.5 puntos por cada formato de la entrega que no se respete.

## Tareas atrasadas

En caso de querer entregar la tarea fuera de plazo, se debe enviar un mail a [msmadrid@uc.cl](mailto:msmadrid@uc.cl) y [ncbarra@uc.cl](mailto:ncbarra@uc.cl) indicando nombre y usuario de Github. Este mail se debe enviar inmediatamente después de subir el último *commit* a revisar.