

Tarea Chica 3: Máquinas de Turing

Ayudantes: Maite Madalosso y Michelle Madrid

Profesor: Denis Parra

11 de Junio de 2021

Indicaciones

- Fecha de Entrega: Viernes 25 de Junio a las 19:59.
 - La entrega de la tarea debe realizarse en el repositorio privado de GitHub Classroom asignado para esta evaluación.
 - 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 \LaTeX .

Introducción [Demon Slayer: DCCinta Infinita]

ADVERTENCIA: Recomendamos encarecidamente leer en enunciado con la siguiente [música de fondo](#).

Tanjiro, Nezuko, Zenitsu e Inosuke han sido llamados para ayudar al mismísimo Alan Turing en la gran batalla contra la Decimotercera (y menos conocida) Luna Demoníaca: "Nyūryoku", demonio que destroza las Máquinas de Turing con *inputs* maliciosos. El problema es que nuestros protagonistas no conocen el funcionamiento de éstas máquinas, por lo que nosotros como expertos programadores les ayudaremos a detener la propagación de estos *inputs* maliciosos por las máquinas de Alan Turing, el Pilar de la Computación (sí, Turing también es un cazador de demonios).

¡Cuidado! La Luna Demoníaca ha detectado nuestra presencia, Zenitsu se ha desmayado y los demás protagonistas, junto a Alan Turing, han comenzado a luchar contra él ¡Pero el demonio sigue lanzando *inputs* maliciosos a las máquinas! Turing necesita que las protejas a como dé lugar y te da una misión especial... la cual será detallada en la sección de Actividades.



Figure 1: Inosuke, Zenitsu, Nezuko, Tanjiro y Alan Turing

Para que Tanjiro, Nezuko, Zenitsu, Inosuke y Alan Turing salgan victoriosos, primero deberás entender mejor el funcionamiento de las Máquinas de Turing.

Máquina de Turing

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
Γ	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 δ 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

Para apoyar a nuestros protagonistas en la lucha contra la Decimotercera Luna Demoníaca *Nyūryoku*, deberás rechazar los inputs malvados construyendo una Máquina de Turing M que acepte el siguiente lenguaje:

$$L = \{ n_i \# n_i - 1 \# n_i - 2 \# n_i - 3 \# \dots \# n_i - k \mid \text{para } 1 \leq i \leq N, 0 \leq k \leq i \}$$

Y tenga como alfabeto de entrada:

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

Es decir, una secuencia consecutiva de números binarios ordenados de forma decreciente y separados por un gato.

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

- La secuencia tendrá al menos tres elementos.
- Ninguna resta dará como resultado un número negativo.
- $n_i \in \{0, 1\}^x$, donde x es una constante que representa el largo de un número binario en la secuencia. Es decir, cada n_i de una secuencia tendrá el mismo largo x (pero este largo puede variar de una secuencia a otra).
- La máquina M puede tener más de una cinta.
- En caso de ser necesario, puedes apoyarte de otros símbolos además de $\{0, 1, \#\}$.

Algunos ejemplos de entradas:

101#100#011#010 → acepta

100110#100101#100100#100011 → acepta

11000#11111#10111#10110 → rechaza porque $11000 \leq 11111$

10000#01111#01101#01100 → rechaza porque $01111 - 1 \neq 01101$

En el siguiente [link](#) podrás encontrar cómo vencer a *Nyūryoku* con mayor rapidez.

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 llamada [AdGuard](#).

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

Se sugiere revisar [esta página](#) para entender como funcionan las operaciones con números binarios. Está permitido usar fuentes adicionales como ayuda, siempre y cuando sean mencionadas 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)

¡Lo conseguiste! Impediste que las máquinas fueran infectadas con los inputs malvados de *Nyūryoku* ¿Cómo lo hiciste? Nuestros héroes, luego de derrotar a la Decimotercera Luna Demoníaca, quieren saber cómo lo haz hecho para informarle a su líder (y en una de esas te unes como cazador de demonios).

Para esto, utilizando la plantilla .tex que se encuentra en tu repositorio, debes escribir un informe en LaTeX que contenga:

1. (1 punto) Una explicación concisa y completa 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 **cómo** lo hace. Se recomienda usar diagramas que complementen la explicación.
2. (1 punto) Una definición formal para Q, q_0, Γ, F .
3. (1 punto) Una explicación detallada de los estados que componen Q . Para cada estado se debe incluir:

- Bajo qué condiciones la máquina pasa a este estado.
- A qué estados puede pasar la máquina.
- Qué 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.

Bonus: Concurso de Memes (2 décimas)

Como ya es usual, aquellos que elaboren y envíen los 3 mejores memes tendrán una bonificación de 2 décimas en esta tarea, las condiciones son las siguientes:

1. Junto a su tarea deben entregar un archivo en formato JPG, PNG o GIF con el nombre **meme_TC3_apellidos.extensión**.
2. Nuestro comité experto en memes tendrá en mente la siguiente rúbrica:
 - Nivel de creatividad.
 - Relación con los contenidos de la tarea.
 - Sentido del humor.
 - Respeto con los compañeros/ayudantes/profesores al usar imágenes de otras personas, recomendamos siempre verificar la autorización para evitar ofensas.
3. Las decisiones del comité no son apelables.

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](#).
- Opcional: archivo con el meme.

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.

Entregas atrasadas

Si así lo desea, existe la posibilidad de entregar la tarea fuera de plazo, con hasta 3 horas de atraso.

Cualquier entrega realizada pasada la hora estipulada de entrega será considerada como atrasada, **sin excepciones**. Cada hora o fracción implicará un descuento de **1.5 puntos** a su nota final.

Debido a lo anterior, recomendamos fuertemente enviar su tarea con anticipación, realizando commits intermedios de ser necesario.

¡Éxito en lo que queda de semestre!



Figure 2: Rengoku comiendo.