



Universidad Austral de Chile
Conocimiento y Naturaleza



Tarea: Máquina de Turing INFO 139 - Teoría de Autómatas.

Profesora: Maria Eliana de la Maza Werner
Instituto de Informática, Universidad Austral de Chile.
Junio 18, 2021

Integrantes

Daniel Ángel Alberto Díaz Vega

Nicol De Lourdes Huaiquil García

Gustavo Alejandro Reyes Romero



Universidad Austral de Chile

Conocimiento y Naturaleza



Introducción.

Una Máquina de Turing es un modelo matemático abstracto de un computador, diseñado por Alan Turing. Este modelo consiste de una cinta semi infinita de celdas que contienen un símbolo de algún lenguaje y un cabezal que lee un símbolo en un determinado estado. Una Máquina de Turing es capaz de reconocer una amplia gama de lenguajes y funciones matemáticas. En este documento se explicará cómo se escribió un programa capaz de simular una Máquina de Turing con la capacidad de reconocer lenguajes.

Desarrollo.

Lo primero que se hizo fue comenzar a realizar el código base en el lenguaje de programación C++, una vez finalizada esta etapa se tuvo que planificar y realizar la parte de la interfaz gráfica, en donde hubo dificultades y por esta razón se decidió cambiar al lenguaje de programación Python, en donde las estructuras de datos se tuvieron que adaptar al nuevo lenguaje, y por eso para poder almacenar el estado inicial, estado final, transiciones y palabra de entrada se utilizaron las estructuras y variables:

Estado inicial: un número entero mayor o igual a 0.

Estado final: un número entero mayor o igual a 0.

Palabra de entrada: un *String*; puede ser vacío.

Transiciones: una función de transición será guardada como un registro, llamado transición, este registro contiene los siguientes atributos:

- *start* : Un entero que guarda el estado actual de la función de transición.
- *startSymbol* : Un carácter que guarda el carácter que se lee en la función.
- *end* : Un entero que guarda el estado al que se cambiara luego de leer el carácter *startSymbol*.
- *endSymbol* : Un carácter que guarda el carácter nuevo que reemplazará al carácter leído en la cinta.
- *move* : Un carácter que indica la dirección en la cual se moverá el cabezal de la cinta. El carácter 'D' significa que el cabezal se moverá a la derecha e 'I' significa que el cabezal se mueve a la izquierda.



Universidad Austral de Chile

Conocimiento y Naturaleza



Todas las transiciones serán guardadas en una tabla hash del tamaño de la cantidad de estados, donde la función de hash está definida como $H(x) = \text{atributo start del registro transición}$.

El programa está compuesto de 3 piezas de código: **workString**, **turing** y **MaquinaTuring**.

- **WorkString** recibe un *string*, en el cual están todas las funciones de transición. Antes de trabajar el *string*, se construye la tabla hash. El *string* es separado por *substrings* que contienen las funciones de transición; de cada *substring* se construyen un registro que contiene la información de la función de transición correspondiente. Luego de construir un registro, este pasa a ser introducido en la tabla hash.
- **Turing** recibe la tabla hash, la palabra, el estado inicial y el estado final. La función inicia dos booleanos para referenciar si la palabra es aceptada o rechazada y la cinta, que guardará cada símbolo de la palabra en una celda. Al introducir cada símbolo, se añade una cantidad de blancos al lado derecho de la palabra. Los blancos son anotados con el símbolo "#". Luego, entra a un bucle *while* que se repite hasta que la palabra haya sido rechazada o aceptada. Dentro de este *while*, habrá una rutina que busca dentro de la tabla hash una función de transición válida con el estado actual y el símbolo que se está leyendo en la cinta actualmente. Si no se encontró una función válida, la palabra el booleano rechaza cambiará a *True* y la palabra será rechazada; en caso contrario, se actualizará el estado actual y el símbolo que se está leyendo en la cinta con el contenido de los atributos *end* y *endSymbol* del registro de la función de transición guardada en la tabla hash; luego, del mismo registro se verifica si el atributo *move* es una "D" para mover el cabezal a la derecha o de lo contrario, mover el cabezal a la izquierda. Al final del bucle *while*, se verificará si el estado actual es un estado final, cambiando el booleano acepta a *True* si es cierto, saliendo del bucle y aceptando la palabra; en el caso opuesto, se sigue iterando en el bucle *while*.



Universidad Austral de Chile

Conocimiento y Naturaleza



- **MaquinaTuring** es el código de la interfaz gráfica del usuario, en el cual se integran las dos piezas de código mencionadas anteriormente en un comando para un botón que verifica con las entradas dadas si la palabra es aceptada por la MT. Se utilizaron dependencias de la librería **tkinter** para la construcción de la interfaz gráfica.

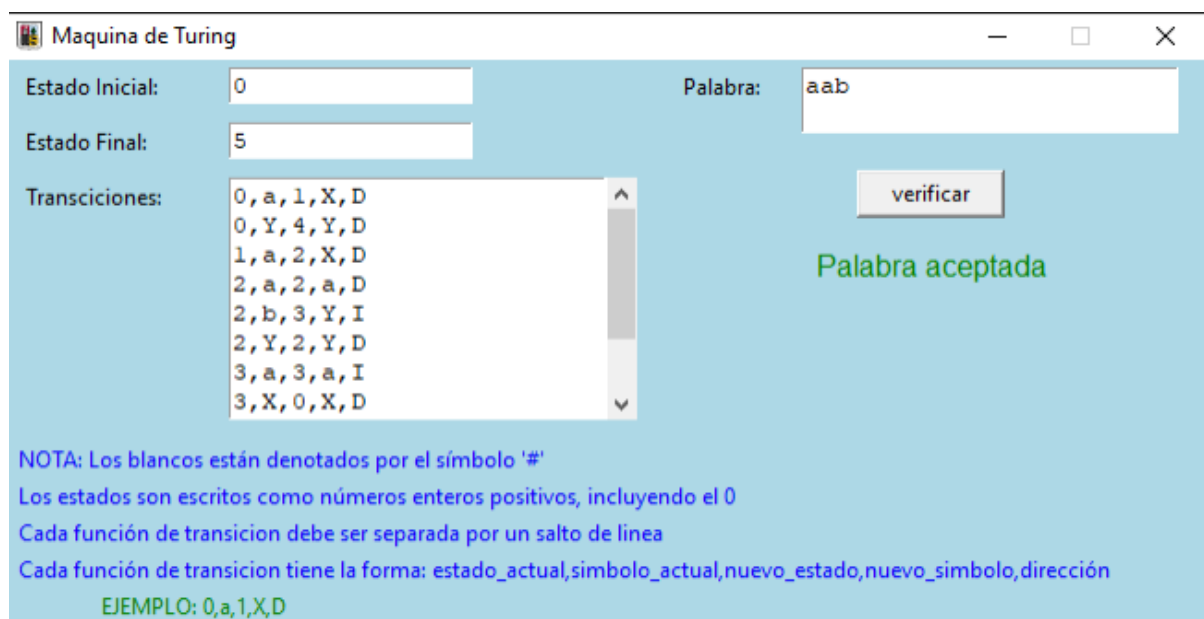
Especificaciones del programa.

- Lenguaje de programación: Python.

Requerimientos de software y hardware:

- Sistema Operativo: Windows 10 de 64 bits.
- Procesador: Intel(R) Pentium(R) N3710.
- Espacio requerido: 10 MB.
- Memoria: 8 GB RAM.
- GPU: Intel(R) Graphics.

Para ejecutar el programa el usuario debe hacer doble clic sobre el archivo ejecutable “MaquinaTuring.exe”. Al ejecutarse, aparecerá una ventana como la siguiente:





Universidad Austral de Chile

Conocimiento y Naturaleza



En los 4 campos de texto se pedirán las entradas de estado inicial, estado final, funciones de transición y palabra. Luego de llenar las entradas con la información necesaria, el usuario deberá presionar el botón verificar para ver si la palabra es aceptada o rechazada; luego de presionar el botón, aparecerá un mensaje bajo el botón que notificará al usuario si la palabra fue aceptada o rechazada por la Máquina de Turing.

Conclusiones.

Según el modelo abstracto de la Máquina de Turing, la cinta es semi infinita, con un número infinito de blancos al lado derecho de la palabra introducida. Este programa no es así, ya que solo tiene un número fijo de blancos que depende del largo de la palabra, en caso de que se quieran utilizar más de los blancos disponibles, el programa fallará. Se consideró la idea de añadir blancos en la cinta a medida que el cabezal se aproxima a los últimos símbolos de la cinta en el lado derecho.

En la interfaz gráfica del usuario, cada vez que se presiona el botón verificar, se llama a la función **workString**, la cual arma la tabla hash de las funciones de transición, esto se debería hacer una sola vez al introducir la MT y a **turing**, la parte del código que simula a una MT; **workString** debería llamarse una sola vez al introducir la MT y pueda guardar la estructura hasta que el usuario quiera introducir otra MT. Se pueden hacer ajustes a la interfaz gráfica del usuario para evitar llamados innecesarios, como un botón adicional que guarde estado inicial, estado final y las funciones de transición al presionarlo una vez, y la próxima vez que se presione, elimine la información introducida en esos 3 campos.