Máquina de Turing

*Fundamentos Teóricos de la Informática Trabajo Integrador*

En este documento se realiza una investigación que es una máquina de Turing, cuáles son sus partes que la componen, para lograr realizar un sistema hecho en Laravel.

Apablaza Karen

25/11/2021

Máquina de Turing

Fundamentos Teóricos de la Informática Trabajo Integrador

Introducción

La máquina de Turing es un autómata finito que dispone de una única cinta de longitud infinita en la que se pueden leer y escribir datos, uno a la vez, usando un cabezal lector/escritor de cinta. Esos datos son símbolos sobre una tira de cinta de acuerdo con una tabla de reglas.

Una máquina de Turing consta de:

* Una **Cinta** que se divide en celdas, una al lado de la otra. Cada celda contiene un símbolo de algún alfabeto. El alfabeto contiene un símbolo especial **llamado espacio en blanco** (Escrito como “B”) y uno o más símbolos adicionales. La cinta se extiende hacia la izquierda o hacia la derecha Las celdas que no se hayan escrito previamente se asumen que están rellenas con el espacio en blanco.
* Un **Cabezal** siempre está situada en una de las casillas de la cinta y mueve la cinta hacia la izquierda y a la derecha (solo una) celda a la vez. Un movimiento de la máquina de Turing puede hacer:
  + Cambiar de estado. El siguiente estado puede ser opcionalmente el mismo que el estado actual.
  + Escribirá un símbolo de cinta en la casilla que señala la cabeza. Este símbolo de cinta reemplaza a cualquier símbolo que estuviera anteriormente en dicha casilla. Opcionalmente, el símbolo escrito puede ser el mismo que el que ya se encontraba allí.
  + Moverá la cabeza de la cinta hacia la izquierda o hacia la derecha. En nuestro caso, exigiremos que haya un movimiento y no permitiremos que la cabeza quede estacionada
* Un **Registro de estado** que almacena el estado de la máquina de Turing, uno de los estados finitos. Hay un estado inicial con el que el registro de estado se inicia.
* Una **Tabla** finita de instrucciones llamada como **Reglas de transición.** En nuestro caso almacenaremos las reglas de transición de la siguiente manera:

(Estado actual) (Valor) | (Nuevo Valor) (Dirección) (Nuevo Estado)

Ejemplo

Vamos a diseñar una máquina de Turing y a ver cómo se comporta con una entrada típica. La máquina de Turing que vamos a construir aceptara el lenguaje {0n1n | n >= 1}.

Se proporciona a la cinta una secuencia finita de ceros y unos, precedida y seguida por secuencias infinitas de espacios en blanco. La máquina cambiará primero un 0 por X y luego un 1 por Y, hasta que se hayan cambiado todos los ceros y los unos.

En la cinta escribiremos lo siguiente:

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 1 | 1 |

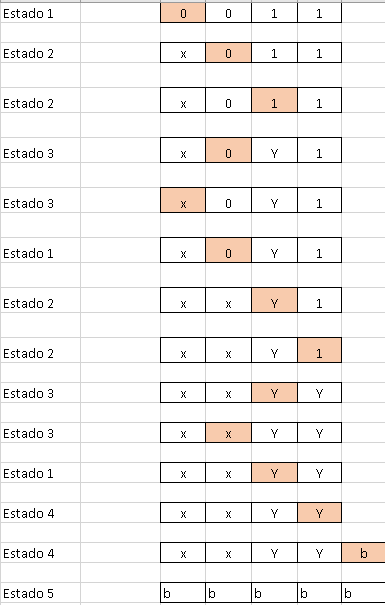
Estado Inicial: 1

Estado final: 5

Símbolo especial o en blanco: b

Tendremos estas reglas de transición:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Estado | Valor | Nuevo Valor | Dirección | Nuevo Estado |
| 1 | 0 | X | Derecha | 2 |
| 1 | Y | Y | Derecha | 4 |
| 2 | 0 | 0 | Derecha | 2 |
| 2 | 1 | Y | Izquierda | 3 |
| 2 | Y | Y | Derecha | 2 |
| 3 | 0 | 0 | Izquierda | 3 |
| 3 | X | X | Derecha | 1 |
| 3 | Y | Y | Izquierda | 3 |
| 4 | Y | Y | Derecha | 4 |
| 4 | B | B | Derecha | 5 |

La secuencia completa de movimientos de la maquina es:

🡪El cabezal lee el primer cero que se ingresó en la cinta, configuramos que empiece del estado 1, busca la condición (estado 1, valor 0), reemplaza en la cinta por una X, se mueve a la derecha y pasa al estado 2.

🡪En el estado 2: Busca la condición (2,0), reemplaza en la cinta por el nuevo valor 0, se mueve el cabezal a la derecha y sigue en el estado 2.

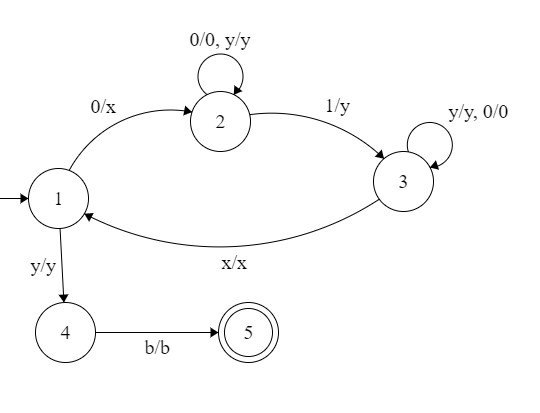
🡪En el estado 2: Busca la condición (2,1), reemplaza con Y, y se mueve hacia la izquierda al estado 3

🡪En el estado 3: Busca la condición (3,0), reemplaza 0, se mueve el cabezal a la izquierda y sigue en el estado 3

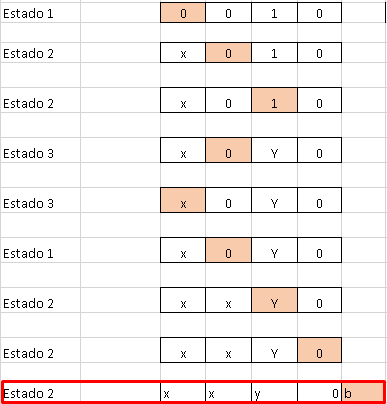
🡪En el estado 3: Busca la condición (3,X), reemplaza con una x se mueve el cabezal a la derecha y pasa al estado 1.

🡪Así sucesivamente hasta que allá puesto en la cinta xxyy, como no hay más ceros y uno, llega un espacio o el símbolo en blanco que definimos en la configuración de la máquina de Turing y se detiene. Entonces la cinta quedo así: XXYY

El diagrama de transición de una máquina de Turing que acepta cadenas de la forma 0n1n. es:



Si cambiamos la cinta e ingresamos 0010, los movimientos son los siguientes:



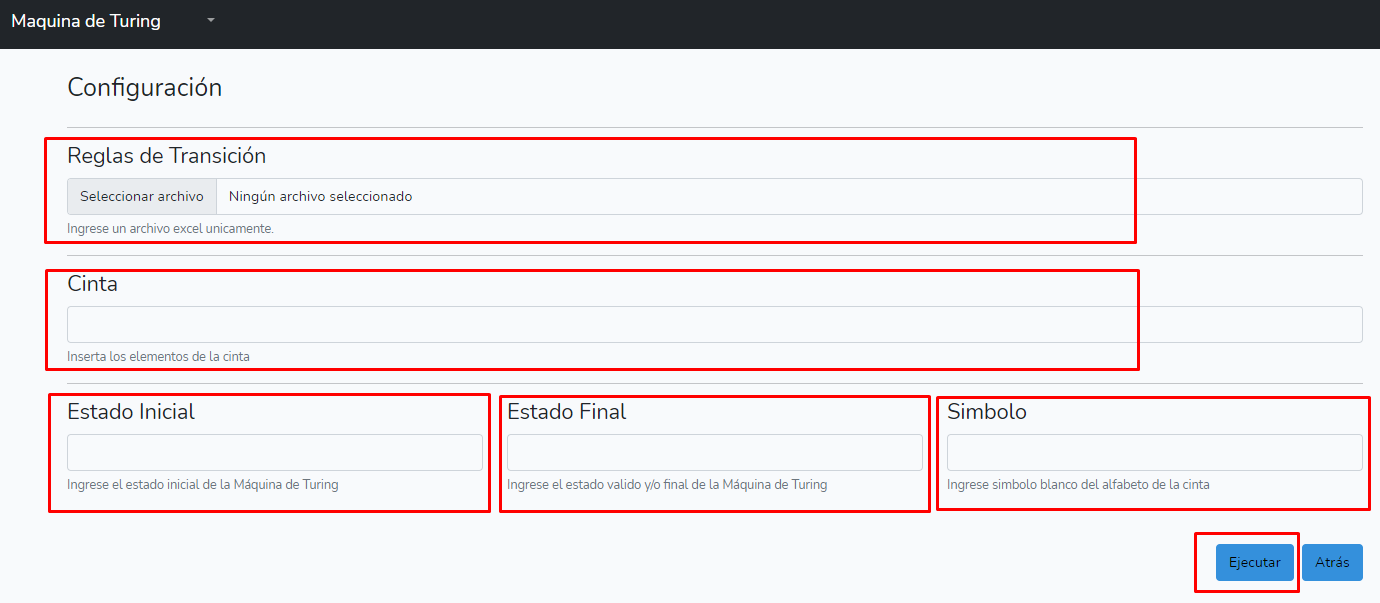
Los movimientos son los mismos que el comportamiento del 0011. Cuando se llega al estado 2 con un 0, lo señala como final por primera vez. La máquina tiene que moverse hacia la derecha donde leerá un símbolo b y el estado 2 cuando llega la condición “b la maquina deja de funcionar y no acepta su entrada.

Programa en laravel

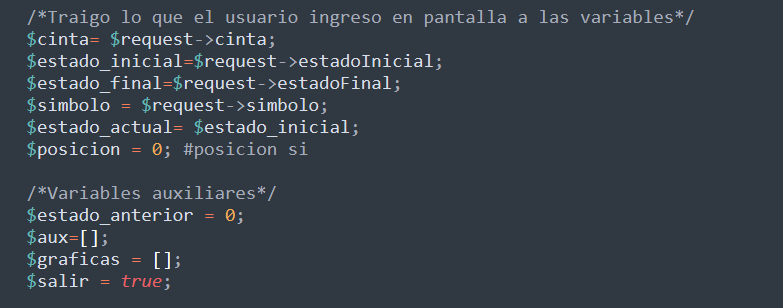
Viendo el ejemplo anterior, implemente lo mismo mostrando los movimientos que realiza el cabezal con la cinta

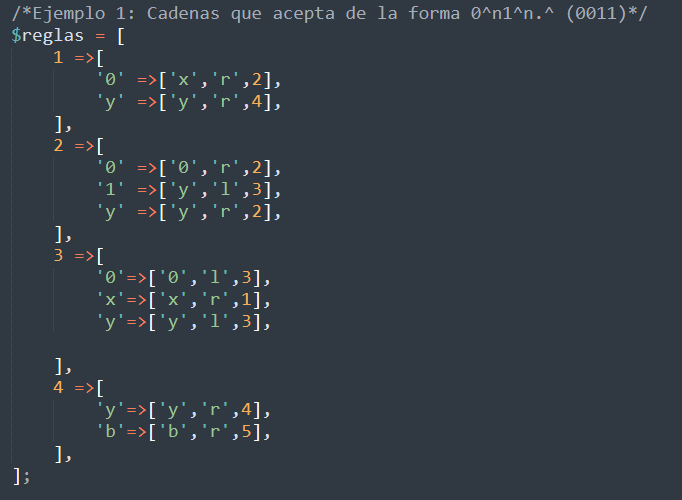
1. Al ingresar al sistema encontraran una pequeña presentación:
2. Deberán apretar el botón de configuración, para poder cargar lo siguiente:
   1. Las **reglas de Transición** en formato Excel (que por el momento no está funcionando, lo cargo manualmente en el programa).
   2. El usuario cargara la **cinta**
   3. **Estado Inicial** que comenzara con el primer elemento de la cinta
   4. **Estado Final** que sería el estado valido y/o final de la máquina de Turing.
   5. El **símbolo blanco** donde se llenará después de lo que ingreso en la cinta o donde terminará dependiendo las reglas de transición

Todo debe coincidir con las reglas de transición por el contrario se producirá algún error. Una vez completado todo presionan el botón Ejecutar.



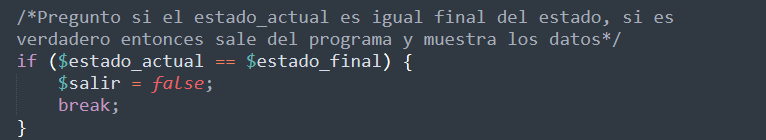
El programa traerá los datos ingresados por el usuario y lo defino en unas variables:

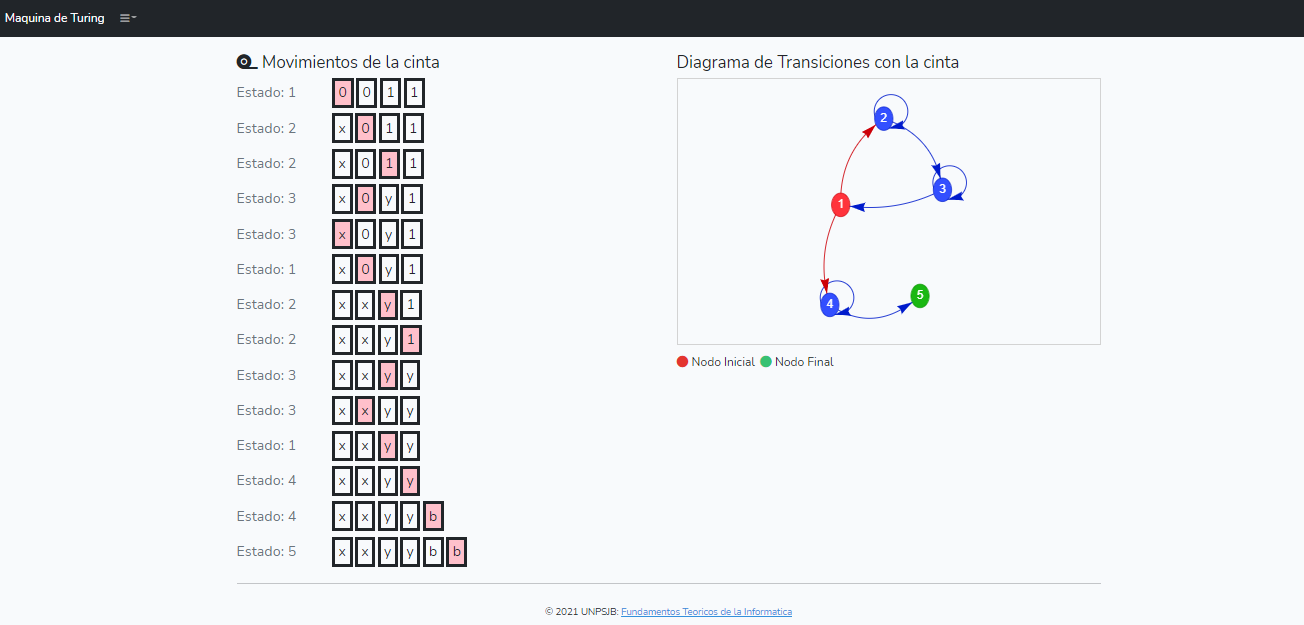


Acá defino mis reglas, por el momento así:

Luego voy recorriendo la cinta y busco en las reglas el elemento de la cinta y muevo el cabezal dependiendo de lo que está definido en las reglas.

Dentro del bucle pregunto si el estado actual es igual al estado final, si es verdadero entonces sale del programa y muestra los datos.



Luego guardo todos los datos en una variable para mostrar en la gráfica en este en las view.

Por el lado izquierdo están los estados con las posiciones del cabezal dentro de la cinta y por el lado derecho un gráfico de mostrando las relaciones entre los estados.

Posibles mejoras

* Que se pueda cargar un Excel con las reglas, sin necesidad de escribirla en el programa
* Más información a la hora de mostrar los datos
* Hacer en tiempo real el movimiento de la cinta.