

PRÁCTICA 9

Implementación de un simulador de Máquinas de Turing

Factor de ponderación: 9

Semana del 13 al 17 de noviembre de 2023

1. Objetivos

El objetivo de la práctica es consolidar los conocimientos adquiridos sobre máquinas de Turing al mismo tiempo que se continúan desarrollando capacidades para diseñar y desarrollar programas orientados a objetos en C++. Mientras que en la anterior práctica se utilizó una herramienta externa (JFLAP [2]) para simular máquinas de Turing, en esta práctica desarrollaremos nosotros mismos un programa que nos permita simular cualquier máquina de Turing de una cinta (y una pista). La máquina de Turing a simular se especificará mediante un fichero de entrada. El programa leerá desde un fichero las características de la máquina de Turing y, a continuación, simulará el comportamiento de la máquina de Turing a partir del contenido de una cinta (también especificada a través de un fichero de entrada).

A la sesión práctica se deberá llevar este simulador implementado y además de validar el funcionamiento del programa también se le pedirá a los estudiantes que pudieran incorporar algún tipo de cambio sobre el comportamiento original del programa. Tal y como venimos insistiendo a lo largo de las prácticas de la asignatura, si el alumnado tiene dudas respecto a cualquiera de estos aspectos, debiera acudir al foro de discusiones de la asignatura para plantearlas allí. Se espera que, a través de ese foro, el alumnado intercambie experiencias y conocimientos, ayudándose mutuamente a resolver dichas dudas. También el profesorado de la asignatura intervendrá en las discusiones que pudieran suscitarse, si fuera necesario.

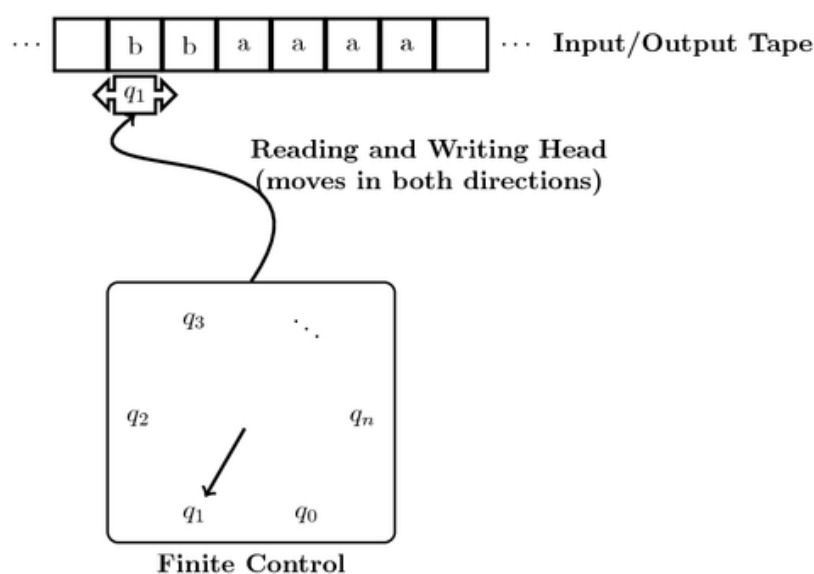


Figura 1: Estructura general de una máquina de Turing

2. Máquinas de Turing

La máquina de Turing [1] dispone de una cabeza que puede leer y escribir símbolos en la cinta de la máquina. Esta cinta se supone que se extiende indefinidamente hacia derecha e izquierda. Una TM utiliza su cinta como dispositivo de almacenamiento auxiliar y puede no sólo insertar o extraer símbolos sino también rastrear los datos de la cinta y modificar las celdas que desee sin alterar el resto. Es conveniente que la TM utilice marcas especiales para distinguir partes de la cinta. Con este fin, la TM utiliza símbolos que no aparecen en los datos de entrada, es decir, se distingue entre el conjunto finito de símbolos llamado el alfabeto de la máquina, en el que deben estar codificados los datos de entrada iniciales y otro conjunto también finito de símbolos de cinta posiblemente mayor que el anterior y al que se denomina alfabeto de la cinta, y que representa al conjunto de los símbolos que la máquina puede leer y escribir. De este modo los símbolos de cinta de una TM pueden incluir marcas especiales que no sean símbolos del alfabeto de la máquina. El espacio en blanco, o simplemente *blanco*, es uno de los símbolos de esta clase: se supone que está en cualquier celda de la cinta que no esté ocupada. Por ejemplo, si una TM leyera más a la derecha de los símbolos de entrada de su cinta, encontraría y leería las celdas en blanco que allí encontraría. Es decir, las celdas que inicialmente no contienen símbolos de la entrada contienen un blanco. También puede ser necesario que una TM borre una celda de la cinta, escribiendo en ella un espacio en blanco. Por estos motivos se considera que el blanco forma parte de los símbolos de cinta, pero no del alfabeto de la máquina. Para que el símbolo de espacio en blanco no nos ocasione problemas en las representaciones, es necesario adoptar algún convenio para representar el símbolo. Por ejemplo, y de forma análoga o como hacíamos para la cadena vacía, utilizaremos $\$$ para la representación de los espacios en blanco. Nótese que JFLAP [2] sigue su propio convenio a la hora de denotar las ϵ – *transiciones* en los NFA así como el *blanco* en las TM.

En cada paso de cómputo, una TM lee un símbolo de la cinta de entrada, lo reemplaza por un nuevo símbolo (que pudiera coincidir con el que encontró originalmente) y, a continuación, cambia su estado (el nuevo estado puede ser el mismo en que se encontraba antes de la operación) y realiza un movimiento de la cabeza de lectura/escritura hacia la izquierda o a la derecha. La acción que en un determinado momento del cómputo realizará la TM dependerá del símbolo (símbolo actual) que esté “visible” en ese momento en la celda (celda actual) en que está posicionada la cabeza de lectura/escritura, y del estado actual del mecanismo de control de la máquina.

Una máquina de Turing se puede definir formalmente como una 7-upla $M \equiv (Q, \Sigma, \Gamma, q_0, \mathfrak{b}, F, \delta)$ donde el significado de cada uno de los elementos es:

- Q es el conjunto de estados (Q finito y $Q \neq \emptyset$)
- Γ es el alfabeto de la cinta
- Σ es el alfabeto de entrada (generalmente $\Sigma \subseteq (\Gamma - \{\mathfrak{b}\})$)
- $q_0 \in Q$ es el estado inicial o de arranque
- $F \subseteq Q$ es el conjunto de estados de aceptación
- $\$ \in \Gamma$ es el símbolo blanco ($\$ \notin \Sigma$)
- δ es la función de transición:
 $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$
 $(q, a) \rightarrow (p, t, X) \quad \text{con } p, q \in Q; \quad a, t \in \Sigma; \quad X \in \{L, R\}$

Tal y como ya se ha mencionado, el alfabeto de entrada generalmente será un subconjunto del de la cinta, pero el símbolo blanco no puede formar parte del alfabeto de entrada. Por su parte, la función de transición transforma pares (q, a) , formados por el estado actual q y el símbolo a que se lee en la posición donde se encuentra la cabeza de lectura/escritura, en ternas (p, t, X) donde p es el estado siguiente, t es el símbolo que se escribe en la cinta y X es un movimiento de la cabeza de lectura/escritura que puede ser a la izquierda (L) o a la derecha (R). Por último, cabe destacar que la función de transición δ de una TM es una función parcial, lo cual quiere decir que no tiene por qué estar definida para todos los pares (q, a) del conjunto de partida.

3. Ejercicio

El objetivo de esta práctica es diseñar en C++ un programa simulador de máquinas de Turing. El programa deberá ser capaz de leer la tabla de transiciones de una máquina de Turing así como el contenido inicial de la cinta de entrada de la máquina. Tras obtener esta información, el programa deberá simular el comportamiento de cada una de las acciones a llevar a cabo por la máquina. Cabe destacar que las máquinas de Turing a simular estarán compuestas por **una única cinta y una única pista**.

La máquina de Turing a simular vendrá especificada en un fichero cuya extensión será `tm` (*Turing Machine*). Todo fichero `.tm` contiene la especificación de una tabla de transiciones de una determinada máquina de Turing. El formato del fichero será el siguiente:

- Línea 1: entero indicando el número de estados de la máquina.
- Línea 2: entero indicando el estado de arranque.
- Línea 3: enteros separados por espacios en blanco indicando qué estados de la máquina son finales o de aceptación.
- Línea 4: entero indicando el número de tuplas que componen la máquina.
- Líneas sucesivas: una tupla en cada línea. Los elementos de cada tupla vendrán separados por espacios en blanco (representados aquí gráficamente por el símbolo `_`). El formato de las tuplas es el siguiente:

`estado_entrada_escribe_movimiento_siguienteestado`

- **estado**: entero (identificador) que representa el estado actual.
- **entrada**: símbolo del alfabeto (de la cinta) que se lee en la cinta.
- **escribe**: símbolo del alfabeto (de la cinta) a escribir en la cinta para dicho estado y entrada.
- **movimiento**: movimiento a realizar por la cabeza de lectura/escritura de la cinta:
 - *L*: izquierda
 - *R*: derecha
 - *S*: parar (no moverse)
- **siguienteestado**: entero (identificador) del siguiente estado al que transita la máquina.

Los símbolos del alfabeto de la cinta podrán ser dígitos o letras minúsculas/mayúsculas. Los estados serán números enteros que comenzarán con el estado inicial igual a cero. El símbolo blanco, se representará por el símbolo dólar \$, aunque en algunas herramientas de simulación como el JFLAP [2], por ejemplo, el símbolo blanco se representa mediante el carácter `□`

A modo de ejemplo, en la Figura 2 se muestra una máquina de Turing que reconoce el lenguaje de las cadenas formadas por una secuencia de *a*'s seguidas por una secuencia de *b*'s, tal que el número de *b*'s es igual al número de *a*'s:

$$L = \{a^n b^n \mid n \geq 0\}$$

En este caso, el alfabeto del lenguaje es $\Sigma = \{a, b\}$ y el alfabeto de la cinta es $\Gamma = \{a, b, X, Y, \$\}$, siendo $\$$ el símbolo blanco (aunque en el diagrama se ha representado por el símbolo \square).

En la figura se incluye también el formato del fichero `.tm` que especifica la máquina de Turing del ejemplo.

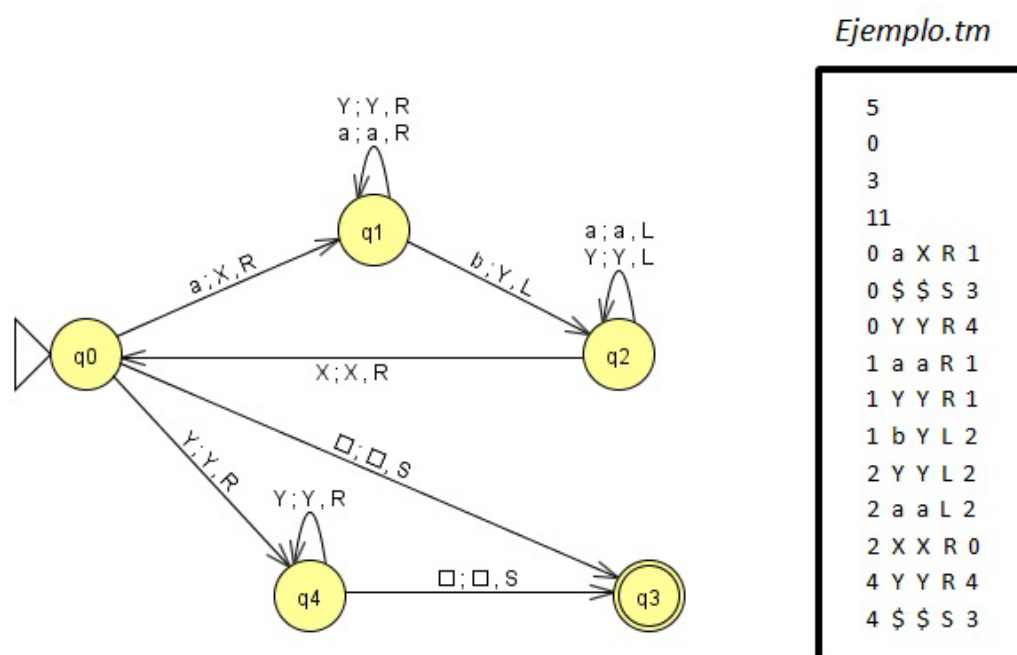


Figura 2: Especificación de una máquina de Turing

El contenido de la cinta de una máquina de Turing vendrá especificado por un fichero cuya extensión será `tape` (cinta). Un fichero `.tape` contiene una única línea en la que se encuentra el contenido actual de la cinta en cuestión. A la izquierda de los caracteres de la cadena especificada en el fichero, la cinta tendrá infinitos símbolos blancos (a pesar de que estos no aparezcan en la propia definición del fichero `.tape`). De la misma forma, a la derecha de los caracteres de la cadena especificada en el fichero, la cinta tendrá infinitos símbolos blancos.

A modo de ejemplo, en la Figura 3 se muestra el contenido de un fichero `.tape` junto con una representación gráfica de cómo se encuentra actualmente la cinta de una determinada máquina de Turing.

Es importante tener en cuenta que en la definición formal de una máquina de Turing, la cinta es un elemento infinito hacia la izquierda e infinito hacia la derecha. A pesar de las limitaciones de los ordenadores en cuanto a disponibilidad de la memoria, hacer un diseño que contemple la opción de que nos podamos mover a izquierda y/o derecha de una cinta, más allá de donde hayamos definido símbolos del alfabeto de entrada.

Ejemplo.tape

aaaabbbb

...	\$	\$	a	a	a	a	b	b	b	b	\$	\$...
-----	----	----	---	---	---	---	---	---	---	---	----	----	-----

Figura 3: Especificación de la cinta

El programa principal recibirá mediante la línea de comandos dos argumentos:

- Nombre del fichero `.tm` donde se encuentra la especificación de la máquina de Turing que se quiere simular. El formato del fichero `.tm` a utilizar seguirá las indicaciones dadas en el guión de esta práctica.
- Nombre del fichero `.tape` donde se encuentra el contenido inicial de la cinta. El formato del fichero `.tape` a generar seguirá las indicaciones dadas en el guión de esta práctica.

Teniendo en cuenta lo anterior, el programa deberá funcionar de la manera siguiente:

1. En primer lugar, se deberá crear una máquina de Turing a partir de la especificación dada en el fichero `.tm` correspondiente. Habrá que notificar al usuario si se produce algún error en la creación (inicialización) de la máquina. A partir de este momento, se le permitirá al usuario mostrar por pantalla la máquina de Turing actualmente definida. Para mostrar la máquina de Turing por pantalla se seguirá el formato establecido para los ficheros `.tm`.
2. El siguiente paso consistirá en la inicialización de la cinta de la máquina de Turing en función de la entrada especificada en el fichero `.tape`. Para iniciar la cinta de la máquina de Turing hay que tener en cuenta que a la izquierda y a la derecha de este contenido inicial tendremos “infinitos” símbolos blancos.
3. A continuación, se procederá a simular el comportamiento de la máquina de Turing partiendo del contenido actual de su cinta. Se deberá simular paso a paso los cambios de estados y modificaciones que se hagan en la cinta hasta que la máquina pare y acepte/rechace la cadena de entrada. Teniendo en cuenta la máquina de Turing presentada en la Figura 2, y suponiendo que la cinta de entrada contiene inicialmente la cadena `aabb`, el comportamiento de la máquina se simulará siguiendo el formato siguiente:

```
$ q0 aabb$
$X q1 abb$
$Xa q1 bb$
$X q2 aYb$
$ q2 XaYb$
$X q0 aYb$
```

```
$XX q1 Yb$  
$XXY q1 b$  
$XX q2 YY$  
$X q2 XYY$  
$XX q0 YY$  
$XXY q4 Y$  
$XXYY q4 $  
$XXYY q3 $
```

Cadena ACEPTADA

Analizando el formato de traza anterior, podemos comprobar que inicialmente, la máquina se encuentra en el estado q_0 y contiene en su cinta la cadena *aabb*, con infinitos símbolos blancos (\$) a izquierda y a derecha: \$ q0 aabb\$

El identificador del estado actual aparecerá siempre justo antes del símbolo sobre el que se encuentra el cabezal de lectura/escritura de la máquina. Teniendo esto en cuenta, en este primer paso, la máquina está en el estado q_0 y sobre su cabezal de lectura/escritura se lee el símbolo 'a' (pues el cabezal de lectura/escritura de la máquina se encuentra sobre la 'a' más a la izquierda de la cinta). Ante esta situación la máquina escribe en esa posición de la cinta un símbolo 'X', moviendo su cabezal de lectura/escritura una posición hacia la derecha (ahora se encontraría sobre la segunda 'a' de la cadena original) y cambiando al estado q_1 como estado actual. Tras este primer movimiento, el estado actual de la cinta sería: \$X q1 abb\$

Hay que tener en cuenta que entre el identificador del estado actual y el contenido de la cinta dejaremos un espacio en blanco a izquierda y a derecha. La traza se mostrará (por pantalla) movimiento a movimiento hasta que la máquina de Turing llegue a un estado para el cual la transición con el símbolo que se encuentre sobre el cabezal de lectura/escritura no esté definida. En ese momento, la máquina “para y acepta” si el estado actual es un estado de aceptación. Si el estado en el que para la máquina es un estado de no aceptación entonces la cadena de entrada será rechazada.

Referencias

- [1] Transparencias del Tema 4 de la asignatura: Máquinas de Turing, <https://campusingenieriaytecnologia2324.ull.es/mod/resource/view.php?id=4936>
- [2] JFLAP: Java Formal Language and Automata Package, <https://www.jflap.org/>