

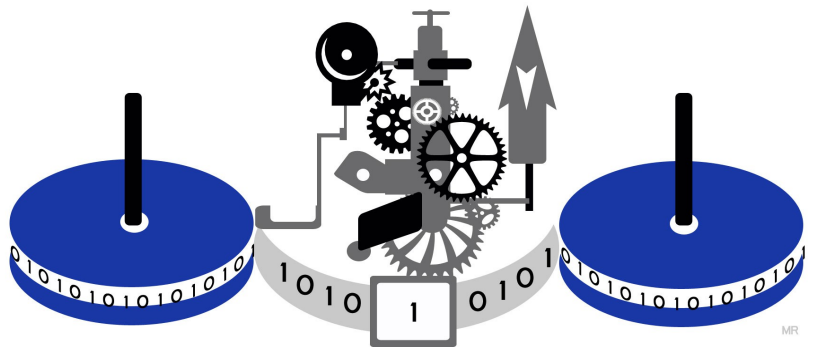
Tarea Programada II – Máquinas de Turing

Introducción

Alan Turing fue un matemático, lógico, criptoanalista, filósofo y pionero de las ciencias de la computación. Entre los muchos aportes que Turing hizo al mundo del saber se encuentra la denominada máquina de Turing. Una máquina de Turing es un dispositivo hipotético que manipula símbolos en una cinta de acuerdo a un conjunto de reglas. A pesar de ser un concepto muy simple, la máquina de Turing puede ser adaptada para simular la lógica de cualquier algoritmo de computadoras y es particularmente útil para explicar las funciones de un CPU. Creada en 1936 y bautizada “*a-machine*” (máquina automática), tuvo como principal intención representar el comportamiento de una computadora y ha servido para ayudar a entender los límites de la computación.

Máquina de Turing

Turing describió la máquina con una capacidad de memoria ilimitada, en forma de una cinta infinita separada en cuadros sobre los cuales puede imprimirse un símbolo. La máquina puede leer un símbolo a la vez y es llamado símbolo leído. Es posible reescribir el símbolo leído, lo cual determina en parte el comportamiento de la máquina, pero en el momento los otros símbolos escritos en la cinta no afectan su comportamiento. Una de las operaciones fundamentales de la máquina es moverse hacia adelante o hacia atrás en la cinta de símbolos, por lo tanto cualquier símbolo en la cinta puede ser procesado eventualmente.



De forma más precisa, una máquina de Turing consta de las siguientes partes:

1. Cinta: se encuentra dividida en celdas que están situadas una al lado de la otra. Cada una contiene un símbolo de un alfabeto finito. El alfabeto contiene un carácter especial “en blanco” y uno o más símbolos. Se asume que puede extenderse indefinidamente hacia la izquierda o hacia la derecha. Las celdas de la cinta que no han sido escritas tienen el carácter en blanco.
2. Cabeza: puede leer y escribir los símbolos que están en la cinta y tiene la capacidad de moverse hacia la izquierda y hacia la derecha, una posición a la vez. También puede quedarse en un mismo lugar.

3. Registro de estados: son los estados que puede tener la máquina, debe ser una cantidad finita. Existe un estado especial que es el estado inicial en el que la máquina se inicializa. También puede contener un conjunto de estados finales o de aceptación.
4. Tabla de instrucciones o función de transición: es el conjunto de instrucciones que indica qué es lo que la máquina debe hacer cuando lee un símbolo de la cinta. Se puede ver como una función que recibe como entrada el estado actual y el símbolo leído, y produce como resultado el estado al que debe moverse, el símbolo que debe escribirse y el movimiento que debe hacer la cabeza, hacia la izquierda, hacia la derecha o quedarse en la misma posición.

De esta forma, en un movimiento de la máquina de Turing, ésta debe:

1. Cambiar de estado, que opcionalmente puede ser el mismo que el actual.
2. Escribir un símbolo en la casilla señalada por la cabeza. Este símbolo sustituye al símbolo que estuviera anteriormente en la casilla. Opcionalmente, el símbolo que se escribe puede ser el mismo que el anterior.
3. Mover la cabeza de la cinta hacia la izquierda, hacia la derecha o quedarse en el mismo lugar.

La descripción formal de una máquina de Turing está dada por una tupla de siete elementos:

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

Donde:

Q: conjunto finito de estados de la máquina.

Σ : conjunto finito de símbolos de entrada.

Γ : conjunto completo de los símbolos de la cinta.

δ : función de transición. Las transiciones de una máquina de Turing pueden representarse visualmente de la misma forma que un grafo etiquetado y dirigido, donde los estados son los vértices y la tabla de instrucciones se representa con los arcos. Un arco que vaya de un estado q a un estado p tendrá una etiqueta formada por tres valores, digamos $X/Y/Z$, donde X representa el símbolo leído, Y el símbolo escrito y Z un movimiento hacia la izquierda ($<$), derecha ($>$) o sin movimiento ($_$).

q_0 : estado inicial en el que se encuentra la máquina el inicial. Es uno de los elementos de Q .

B : símbolo que representa al espacio en blanco.

F : conjunto de estados finales o de aceptación.

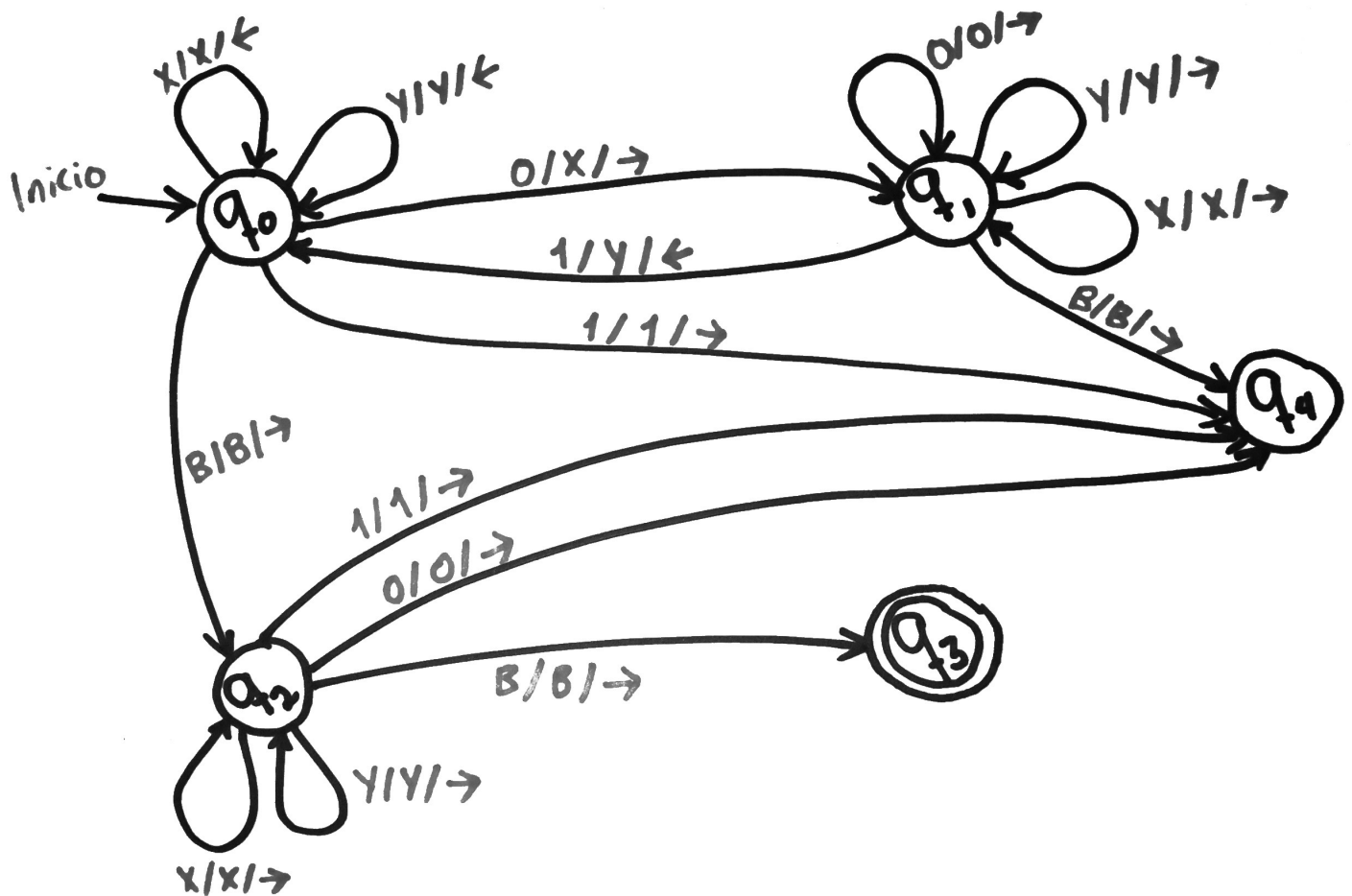
Ejemplo: Supongamos que se desea crear una máquina de Turing que indique si una cadena de dígitos binarios es de la forma $0^n 1^n$, es decir, una cantidad n de ceros seguido de la misma cantidad de unos.

$$Q = \{q_0, q_1, q_2, q_3, q_4\}$$

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

$\Gamma = \{0, 1, X, Y, B\}$

$\delta =$



$q_0 = q_0$

$B = B$

$F = \{q_3\}$

La estrategia de esta máquina es buscar los ceros y sustituirlos con el símbolo X, y los unos y sustituirlos por Y. El estado q_0 es el inicial, indica que se va a buscar un 0 de derecha a izquierda, cuando encuentra uno lo sustituye con X y se mueve hacia la derecha. El estado q_1 busca hacia la derecha un 1, cuando lo encuentra lo sustituye con Y y se mueve hacia la izquierda. El estado q_2 indica que ya fueron sustituidos todos los 0 y se encontró un carácter en blanco, y lo que hace es avanzar hacia la derecha hasta encontrarse otro carácter blanco. El estado q_3 es el estado de aceptación, la máquina termina su ejecución. El estado q_4 es un estado de error en el que la máquina queda en caso de que se lea un carácter no esperado en cualquiera de los estados de la máquina.

Para ejecutar la máquina es necesario indicar la cadena de símbolos de entrada y situarla en el estado inicial. Por ejemplo, al darle como entrada la tira "000111" la máquina llega eventualmente al estado q_3 . Pero si se le da la tira "00011" o "00111" la máquina llega al estado de error q_4 .

Software a desarrollar

Su trabajo consiste en elaborar un programa en C++ que utilice estructuras de datos para simular gráficamente el funcionamiento de una máquina de Turing cuya definición se encuentre en un archivo de texto.

Primero que todo, el usuario indica el nombre del archivo de texto donde se encuentra la especificación de la máquina y el programa debe leerlo para representarla en memoria. El formato del archivo es simple, en cada línea se especifican las diferentes partes de la máquina, cada línea empieza con un carácter que indica qué es lo que viene en dicha línea: Q para estados, S para símbolos de entrada, G para símbolos de la máquina, T para transición, I para indicar el estado inicial, B para el carácter en blanco, F para estados de aceptación, R para estados de rechazo. Las transiciones se representan una por línea, con cinco elementos cada una: estado origen, carácter leído, estado destino, carácter escrito, movimiento de la cabeza. El archivo de texto asociado a la máquina anterior contendría las siguientes líneas:

Q:q0,q1,q2,q3,q4

S:0,1

G:0,1,X,Y,B

T:q0,X,q0,X,<

T:q0,Y,q0,Y,<

T:q0,0,q1,X,>

T:q0,1,q4,1,>

T:q0,B,q2,B,>

T:q1,0,q1,0,>

T:q1,1,q0,Y,<

T:q1,X,q1,X,>

T:q1,Y,q1,Y,>

T:q1,B,q4,B,>

T:q2,0,q4,0,>

T:q2,1,q4,1,>

T:q2,X,q2,X,>

T:q2,Y,q2,Y,>

T:q2,B,q3,B,>

I:q0

B:B

F:q3

R:q4

Nótese que para efectos de la implementación puede ser que algunos de los datos de la especificación formal de la máquina no sean completamente necesarios, como sucede con el símbolo en blanco. También, se está agregando un tipo de estado de rechazo, en el cual la máquina se debe detener si llega a alcanzarlo.

Cuando el programa lee el archivo, debe crearse una estructura de grafo en memoria que represente los estados y las transiciones de la máquina. Entre las decisiones de diseño que deben tomarse está el tipo de representación que se va a utilizar para el grafo (matriz o listas de adyacencia).

Una vez creada la estructura, todos los datos relevantes de la máquina deben presentarse en pantalla. Se requiere que la representación sea en modo gráfico y no en consola.

Seguidamente el usuario debe solicitar al usuario una tira de entrada para ejecutar la máquina. El programa debe hacer las validaciones necesarias para que esta tira contenga sólo caracteres especificados en el conjunto de símbolos de entrada.

Una vez validada la tira de entrada, ésta también debe presentarse en pantalla y ubicar la posición de la cabeza de la máquina en el primer símbolo de la tira. El usuario indica al programa que ejecute “un paso” en la máquina, lo que significa que la máquina debe leer el símbolo actual, buscar la transición correspondiente, pasar al estado indicado en la transición, escribir el símbolo correspondiente y mover la cabeza en la dirección indicada. Todos los cambios deben mostrarse en pantalla.

El usuario puede ir ejecutando paso a paso el funcionamiento de la máquina e ir observando el cambio que va generando en la tira.

Cuando la máquina llega a un estado de aceptación o rechazo, debe indicarlo en pantalla.

En cualquier momento de la ejecución, el usuario puede abortarla e indicar si desea cargar otro archivo o si desea ejecutar nuevamente la máquina.

Recomendaciones para la implementación

Es importante que se abstraigan las diferentes funcionalidades que se necesitan en el proyecto, para programarlas separadamente y poder reutilizarlas. A continuación se muestran algunas funcionalidades que podrían abstraerse:

- Leer archivo y cargar en la estructura de grafo
 - Lectura de archivos de texto línea por línea
 - Separación de un string en diferentes elementos
 - Creación de estados
 - Creación de transiciones
- Dibujar en pantalla los estados y transiciones de la máquinas
Para este punto se recomienda representar los estados siempre en una forma organización circular para que sea posible unir cualquier par de estados por medio de líneas rectas.
 - Representación de los estados
 - Representación de las transiciones
- Dibujar en pantalla los contenidos de la tira y la posición de la cabeza
- Ejecutar un paso de la máquina
 - Obtener símbolo actual
 - Cambiar el estado actual de la máquina
 - Escribir un símbolo en la tira
 - Mover la cabeza

Analice estas funcionalidades e identifique qué entradas, salidas y restricciones tienen cada una de ellas. Identifique otras funcionalidades que le puedan ayudar a cumplir el objetivo de programar el proyecto.

Documentación

Como documentación interna, debe indicar lo siguiente:

1. Por cada archivo fuente: Autor(es) y fecha de creación.
2. Clases: descripción breve de la clase y su objetivo.
3. Métodos: objetivo del método, entradas y salidas.
4. Otros: cualquier otro comentario que crea necesario para aclarar el funcionamiento de alguna porción de código.

Escriba código claro y conciso, trate de apegarse a los principios de código limpio para el código que escriba. Recuerde que es mucho más importante que su código sea claro y fácil de entender que si es muy eficiente. Evite los comentarios excesivos.

En cuanto a la documentación externa, debe entregarse un documento en formato PDF con las siguientes secciones:

- a. Portada
- b. Introducción. ¿Por qué se hace el proyecto y qué se incluye? (1 págs.)
- c. Presentación y análisis del problema (4+ págs.)
 - i. Qué es lo que hay que resolver. Identificar pequeños problemas que deben resolverse en el proyecto.
 - ii. Cómo se va resolver el problema. La forma en que se planea resolver el problema.
 - iii. Análisis crítico de la implementación. Luego de la implementación, decir qué se logró implementar, lo que faltó y qué cosas se podrían mejorar de lo que se implementó. No sólo mencionar, si no explicar por qué.
- d. Conclusiones: resoluciones puntuales tras el proyecto. Estas deben ser relacionadas con los aspectos técnicos del trabajo únicamente. (1-2 págs.)
- e. Recomendaciones: consejos o advertencias que se derivan de las conclusiones. Lecciones aprendidas durante el desarrollo del proyecto. Recomendaciones para personas que tengan que hacer el mismo trabajo. También deben estar orientadas con aspectos técnicos de la tarea programada. Se recomienda hacer una o más recomendaciones por cada conclusión. (1-2 págs.)
- f. Referencias. Deben incluirse en formato APA.

Forma de trabajo

El proyecto se desarrollará en tríos o parejas. No se permitirá la entrega de proyectos desarrollados de forma individual a menos que exista autorización previa del profesor.

En la medida de lo posible, presente al menos un adelanto de trabajo al profesor o asistentes del curso durante la primera semana y media del proyecto. Esta supervisión le puede ayudar a mejorar el resultado final del proyecto. No dude en consultar cualquier asunto tanto de programación como de elaboración de la documentación.

Los miembros del grupo deben distribuir el trabajo uniformemente, cualquier situación en la que un miembro no esté realizando su parte del trabajo debe ser notificada al profesor inmediatamente y dicha persona puede ser separada del grupo para realizar el trabajo de forma individual.

Entrega

El tiempo asignado para la tarea programada es de 3 semanas.

Evaluación

La tarea tiene un valor de 20% de la nota final, en el rubro de Proyectos Programados.

Desglose de la evaluación de la tarea programada:

Documentación: 15%

Programación: 85%

Recomendaciones adicionales

Pruebe cada funcionalidad individualmente. No implemente grandes secciones del programa sin verificar el funcionamiento por separado de cada una de sus partes. Esto dirige a errores que son más difíciles de encontrar.

Recuerde que el trabajo es en equipos, es indispensable la comunicación y la coordinación entre los miembros del subgrupo.

Comparta el conocimiento con los demás compañeros de grupo y de la carrera, la ciencia de la computación es una disciplina que requiere el traspaso libre de conocimientos. Se logran mejores resultados con la colaboración de todos que con el esfuerzo separado de diferentes personas.

No dude en consultar diferentes fuentes para satisfacer las dudas. Aparte de las búsquedas en Internet, asegúrese de exponer sus dudas a sus compañeros, profesor y conocidos que estudien la carrera; en la mayoría de las ocasiones es más provechosa conversación de 10 minutos entre personas que están trabajando en lo mismo que pasar horas buscando la respuesta a una duda de forma individual.

No deje la documentación para el final, es buena práctica ir desarrollándola durante todo el transcurso del proyecto. Recuerde que la documentación debe ser concisa y puntual, por lo que en realidad no toma mucho tiempo al realizarla de esta forma.

Plagios no serán tolerados bajo ninguna circunstancia. Cualquier intento de fraude será evaluado con una nota de cero y se enviará una carta al expediente del estudiante. Siempre escriba su propio código.

Se recomienda la utilización de la biblioteca WinBGIm para la elaboración de la interfaz, debido a su simplicidad de uso. Si se va a utilizar otro tipo de biblioteca debe comunicarlo al profesor y proveer todos los archivos necesarios para su ejecución.

Referencias

Hopcroft, John E., Rajeev Motwani, and Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. 2da ed. Madrid: Pearson Educación, 2002.

Alan Turing. (2014, October 19). In *Wikipedia, The Free Encyclopedia*. Retrieved 02:20, October 20, 2014, from http://en.wikipedia.org/w/index.php?title=Alan_Turing&oldid=630216215

Turing machine. (2014, October 11). In *Wikipedia, The Free Encyclopedia*. Retrieved 02:12, October 20, 2014, from http://en.wikipedia.org/w/index.php?title=Turing_machine&oldid=629186967

Imagen de portada:

Turing Machine by Matt Rolph. In *Portfolio Blog*. Retrieved 20:30, October 10, 2014, from <http://design.vidanto.com/?p=225>