

4.º CONGRESO
INFORMATICA Y AUTOMATICA
MADRID -16/19 OCTUBRE 1979

SESION

11

SIMULADOR DE MAQUINAS DE TURING PARA
FINES DIDACTICOS

por

C. Delgado Kloos y F. Sáez Vacas
Laboratorio de Ordenadores, Cibernética y Teoría de Sistemas.
E.T.S.I. de Telecomunicación. (U.P.M.)

RESUMEN

Se presenta un programa simulador escrito para ayudar a los estudiantes a comprender el concepto de máquina de Turing y a probar y corregir sus diseños de algoritmos con esta máquina.

En una primera parte se describe el simulador desde el punto de vista de su uso, en el que cabe resaltar la representación del esquema funcional en la forma de quintuplas, junto a una serie de opciones de explotación.

La segunda y última parte se dedica a las características del programa, ocupación de memoria, representación y manejo de estados internos, simulación de la cinta y otras técnicas - propias del programa o del miniordenador en el que aquel ha sido instrumentado.

1. INTRODUCCIÓN. OBJETIVOS DEL SIMULADOR.

Un estudiante no tiene dificultad alguna en comprender la definición de una máquina de Turing como una séxtupla

$$\langle S, Q, D, f, g, h \rangle$$

S = conjunto finito de símbolos en la cinta (alfabeto externo).

Q = conjunto finito de estados del control (alfabeto interno).

D = conjunto finito de desplazamientos de la cabeza (+, =, -).

f = función de símbolo (siguiente)

g = función de estado (siguiente)

h = función de desplazamiento (siguiente)

y el principio de funcionamiento de este autómata que con una cabeza de lectura/escritura lee un símbolo en una cinta y dependiendo del estado de su elemento de control, escribe un símbolo u otro sobre la misma posición de la cinta, desplazando luego la cabeza (también se podría desplazar la cinta) una posición hacia la derecha o la izquierda o manteniéndola en el mismo lugar y pudiendo modificar el estado.

Pero la máquina de Turing es, potencialmente, un objeto muy complejo. Por ello mismo el estudiante experimenta gran dificultad en comprender a la máquina de Turing como ejecutora de un algoritmo. Sigue penosamente el razonamiento por el cual, por ejemplo, un determinado esquema calcula el m.c.d. de dos números naturales expresados en forma unaria (véanse cuadros al final).

El programa simulador que se describe en este escrito tiene una finalidad prioritariamente didáctica. Se ha concebido como una ayuda al estudiante para que pueda salvar ese foso y llegar a los aspectos más interesantes de la máquina de Turing. Al estudiante avanzado le servirá como herramienta de trabajo para comprobar y poner a punto sus propios diseños, función clásica de las técnicas de simulación.

Anteriormente se han realizado otros simuladores (por ejemplo: Curtis 1965) con otras finalidades distintas, normalmente relacionadas con problemas de investigación en teoría de algoritmos y calculabilidad. Curtis (1) emplea un lenguaje que permite un modo de descripción de algoritmos para máquinas de Turing alejado de la representación por quintuplas, representación que, en nuestro caso es esencial ya que se trata de potenciar su comprensión en los estudiantes.

En un campo de tan intensa evolución tecnológica como la informática se hace aún más imprescindible mantener unos cuantos puntos de referencia, lo que en metodología formativa equivale a echar mano del principio genético. La máquina de Turing es uno de esos puntos y decidir que profundizar en ella es materia reservada a los estudiantes de informática teórica, la convierte ipso facto a los ojos de los demás en pieza de museo, anacronismo, en curiosidad histórica o, en el mejor de los casos, en una pérdida de tiempo. En nuestra experiencia le atribuimos también una doble dimensión educativa y de test de capacidades lógicas para estudiantes de Informática, que para ser debidamente aprovechada requiere el concurso de ayudas automáticas del estilo de la que aquí se presenta.

2. DESCRIPCIÓN FUNCIONAL (O EXTERNA) DEL SIMULADOR.

Para la realización práctica del simulador se ha empleado el miniordenador SPC-16/65 de "General Automation" de 32 K-palabras, teleimpresor modelo ASR 33 de "Teletype Corporation" e impresora rápida de líneas modelo V-132-C de "Data Printer Corporation". También se utilizó una lectora y una perforadora de cinta de papel del mismo sistema SPC-16/65 para almacenar y leer la lista de quintuplas o esquema funcional lo cual quita al operador el trabajo de teclearlo si utiliza una misma máquina de Turing repetidas veces. (Sobre el convenio seguido de representación del esquema funcional, véase (2)).

La comunicación del programa con el operador es de tipo conversacional. Por programa se van solicitando a través del teletipo los datos que tiene que introducir el operador. Primeramente ha de introducir el número de estados y el de símbolos, después el símbolo que hará de 'blanco' en la -

cinta y a continuación el resto de los símbolos. El primer estado es Q_0 y puede haber en total hasta el Q_{99} . Los símbolos del alfabeto externo pueden ser: !, ", #, \$, %, &, ', - , (,), *, +, ,, -, ., /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, :, ;, <, =, >, ?, @, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, [,], ^.

La introducción del esquema funcional se hace a medida que el programa va imprimiendo por el teletipo las parejas-entrada (símbolo-estado). El operador contesta con la terna-salida, correspondiente de acuerdo con ciertas reglas. El símbolo-entrada es impreso en la octava posición de la línea y el estado a partir de la 16. A esto responde el operador tecleando el movimiento, de acuerdo con un convenio, en la primera posición de la línea siguiente, el símbolo-salida, en la octava y el estado-salida a partir de la 16. El convenio de movimiento es:

- > : la cabeza se desplaza hacia la derecha
- = : la cabeza no se desplaza
- < : la cabeza se desplaza hacia la izquierda

El símbolo tecleado ha de ser del alfabeto externo - definido previamente y el estado puede ser del alfabeto interno o STOP. Si alguno de los elementos de la terna-salida no es correcto, el programa lo advierte y da al operador la oportunidad de repetir la terna. Las ternas de las quintuplas cuyas parejas no pueden producirse, por diseño del esquema funcional, se dejan en blanco.

La impresión del esquema funcional por la impresora rápida es opcional. La lista de quintuplas se puede introducir también por otro periférico que no sea el teletipo, es decir, el esquema funcional puede ser leído de una cinta de papel en la que se almacenó, mientras el resto de los datos (condiciones de impresión, condiciones iniciales) se siguen introduciendo por el teletipo.

Con las condiciones de impresión se especifican los momentos en los que ha de ser impreso el contenido de la cinta. Hay diversas maneras de especificarlos. Se puede dar la

orden de impresión 1) cada cierto número de pasos, 2) cada vez que se cambie de estado I) durante toda la computación o II) en un intervalo a fijar y 3) cada vez que se adopte un estado fijado por el operador. Las situaciones inicial y final aparecen impresas siempre.

Las condiciones iniciales determinan el problema concreto que ha de resolver la máquina de Turing. Son: 1) la información inicial sobre la cinta (que no puede exceder de 1024 símbolos y ha de introducirse por el teletipo en grupos de 71 caracteres, es decir, se ha de pulsar la tecla de retorno de carro cuando el teletipo salte de línea automáticamente), 2) la posición inicial de la cabeza (que se indica con relación a la posición del primer carácter de la información inicial teclada y ha de corresponder a alguno de los elementos del vector que simula la cinta) y 3) el estado inicial (que ha de pertenecer al alfabeto interno).

En la impresora rápida de líneas quedan impresos ambos alfabetos siempre, el esquema funcional, si se quiso, y las opciones de escritura, si se deseó alguna. Las descripciones instantáneas de la computación de la máquina de Turing contienen los siguientes elementos: 1) número de pasos realizados, 2) estado del autómata, 3) contenido de la cinta y 4) indicadores de posición de la cabeza (^) y de dónde se encontraba el primer símbolo de la información inicial (').

3. DESCRIPCIÓN ESTRUCTURAL (O INTERNA) DEL SIMULADOR.

El programa se ha escrito en Fortran IV con las particularidades propias del sistema SPC-16 y su sistema operativo en tiempo real RTOS. El Fortran IV del SPC-16 tiene la posibilidad de definir bytes y bits como variables. Las variables byte se manejan como enteros no pudiendo ser negativos, pero ocupan sólo media palabra, o sea, 8 bits. Las variables bit son variables lógicas cuya información está contenida en un solo bit. El programa ocupa en memoria menos de 13 K-palabras, lo cual le confiere una buena transportabilidad a otras instalaciones.

El programa simulador se puede asimilar a un programa supervisor que se encarga de la compilación y ejecución de un programa Turing.

A continuación se explican algunas técnicas utilizadas a lo largo del programa.

El conjunto de estados siguientes correspondientes a cada pareja-entrada (símbolo-estado) posible es almacenado en una matriz bidimensional cuyos elementos son bytes. Los índices de esa matriz se obtienen biunívocamente a partir de la pareja-entrada. Existe otra matriz de análogas características que alberga el símbolo siguiente de las quintuplas. El movimiento se almacena en dos matrices-bit. Para un par de índices se obtienen dos bits que dan lugar a las cuatro combinaciones: II (Parada instantánea), IO (Parada definitiva), OI (Movimiento a la izquierda), OO (Movimiento a la derecha).

El operador introduce como datos estados como Q0,Q1, Q2..... y símbolos tales como A, B,*..... y espera que el ordenador se comunique con él también utilizando esta nomenclatura. Pero internamente en el programa un estado y un símbolo deben dar lugar a un par de índices que indiquen unos elementos dentro de las matrices que definen el comportamiento de la máquina de Turing. Estos índices deben ser enteros positivos de valor mayor o igual que 1. La conversión en el caso de los estados se produce sumando una unidad al número de orden. En el caso de los símbolos se tiene en cuenta la representación interna de los caracteres alfanuméricos almacenados en variables byte. El conjunto {!, ", #,... [,] ,+} es equivalente internamente al {161, 162, 163,.... 219, 221,222}. Por tanto para pasar al conjunto de índices {1, 2, 3, ...59, 61, 62} basta con restar 160 de la representación numérica del carácter.

La cinta de la máquina de Turing es simulada por un vector-byte de 2048 elementos, que es suficiente para el uso a que se destina el programa simulador. Pero para aprovechar mejor esta zona de memoria se le ha dado por programa a la cinta el carácter de banda continua. Es decir, si se ha llegado al extremo superior del vector y hay que escribir en un

elemento situado más a la derecha, mientras que las posiciones inferiores contienen blancos, se continuará escribiendo en las posiciones bajas del vector. Existen además dos variables enteras, que contienen las posiciones extremas de la información contenida en la cinta, y que sirven para controlar el desbordamiento.

4. OTRAS APLICACIONES DEL SIMULADOR. SUGERENCIAS,

Si al alumno le cuesta trabajo, como ya se ha señalado, deducir el modo de funcionamiento de una máquina de Turing a partir de su esquema funcional, definir un algoritmo en forma de esquema funcional de M. Turing (en otras palabras, diseñar una M. T. para una computación concreta) es cosa que queda para los alumnos más avanzados. Estos encontrarán en el simulador una ayuda inapreciable para confirmar sus diseños y, en definitiva, entrenarse en esta tarea, supuesto que aquellos no sobrepasen los límites del simulador.

Siempre en el terreno educativo, lo hemos utilizado para simular ejercicios de examen y comprobar si ciertas soluciones eran o no correctas y, en su caso, podérselo presentar visualmente a algunos alumnos reticentes. Esta es una aplicación menor, obviamente, pero resulta muy eficaz debido a la fatiga que produce seguir mentalmente el funcionamiento completo de muchas M.T.'s diferentes, una tras otra.

Es sabido que alguna pereza mental nos conduce habitualmente a no comprobar y tomar por buenos planteamientos o soluciones publicados aquí y allá que, de esta manera, se propagan a veces indefinidamente. También en esto el simulador puede ayudar a superar esa pereza. Algunos ejemplos de esto y más detalles sobre el simulador pueden encontrarse en (3) y, especialmente, en (4) donde se desarrollan y discuten organigramas, programas y toda clase de pormenores.

5. REFERENCIAS

1. CURTIS, M.W. A Turing machine simulator. Journal of the A.C.M. Vol. 12, n° 1. En. 1965. pp. 1-13.

2. FERNANDEZ, G. y SAEZ VACAS, F. Fundamentos de los Ordenadores. Vol. II: Elementos de Metateoría, Tema III, E.T.S.I. Telecomunicación, Madrid, 1978. 1ª. edic.
3. DELGADO KLOOS, C. y SAEZ VACAS, F. Simulador de máquinas - de Turing, concebido como instrumento de enseñanza. Laboratorio de Ordenadores, Cibernética y Teoría de Sistemas, - UPM/ETSITM/LOCTS/04-79, Abril 1979.
4. DELGADO KLOOS, C. Simulador de máquina de Turing. Proyecto Fin de Carrera, E.T.S.I. Telecomunicación, Madrid, Septiembre, 1978.