

Programa que encuentra MT óptima HELP

Objetivo

El objetivo de este programa es aproximar la máquina de Turing más pequeña capaz de reproducir una cadena dada. Para la aproximación se utilizará el método de escaladores, específicamente el Random Mutation.

Motivación

La teoría algorítmica de la información está basada en la complejidad de Kolmogorov la cual indica que la complejidad de un programa es igual a la máquina de Turing más pequeña capaz de replicarlo. La dificultad está en que la complejidad de Kolmogorov es incomputable ya que presenta el problema de paro. Sin embargo, sí es posible aproximarlos con varios métodos, por ejemplo, por medio de escaladores.

Descripción

El programa consta de 5 partes principales que se ejecutan en el orden presentado a continuación:

- Tomar la cadena dada por el usuario, convertirla en su respectiva representación en código ASCII binario y escribirla sobre la cinta objetivo. Esto se hace en el método `set_cadObj` (String cad).
- Generar o Mutar la máquina de Turing para lo cual se tienen dos métodos:
 - o `generaMT()`: crea dos tablas de 64x8 y las llena con 1's o 0's aleatoriamente para mandarla al simulador la primera vez que se ejecute el programa.

Observaciones:

1. Una tabla es para el caso en el que la cabeza lectora lea un '0' (`tablaEdos0`) y otra es para la lectura de un '1' en la cinta (`tablaEdos1`).
 2. Cada fila es un estado cuyo número es el número de fila empezando desde cero.
 3. La primera columna se refiere al símbolo escrito, la segunda es la dirección de desplazamiento y las siguientes seis columnas indican el estado destino.
 4. El estado 63 se refiere al estado de HALT.
- o `alteraMT()`: después de crear las tablas la primera vez, se elige de manera aleatoria un bit para hacer la mutación. Para esto el método elige una tabla, un estado y una columna para negar el bit, es decir, si la celda elegida tenía un 1 lo borra y escribe un cero y viceversa.

- Simular la máquina de Turing generada o mutada mediante el método simular ().

La simulación de la máquina de Turing binaria que se utiliza cumple con las siguientes restricciones:

1. Longitud máxima de la cinta ($L_{max}= 10000000$).
 2. Número máximo de estados ($S_{max}=64$).
 3. Número máximo de pasos. ($Pasos_{max}= 500000000$)
 4. Posición inicial de la cabeza lectora en la cinta ($PosInicial=L_{max}/2+1$).
 5. Posición final del resultado a la mitad de la cinta.
 6. La cinta está llena de ceros inicialmente.
 7. El último estado (63) es el estado de HALT (estado de paro).
- Asignar un fitness a la máquina simulada. La métrica para esto es el número de bits diferentes de la cinta obtenida con la máquina de Turing generada o mutada y la cinta objetivo. Por lo tanto, entre más pequeño sea el fitness se estará más cerca del resultado. Si el fitness es mejor que la de la anterior éste se guarda junto con sus respectivas tablas (mejortablaEdos0 y mejortablaEdos1).
 - Una vez que se hayan hecho el número máximo de simulaciones (numSimMax), se toman las mejores tablas encontradas y se simula de nuevo pero fijándonos en los estados realmente visitados para sólo guardar éstos en dos nuevas tablas que representen la aproximación de la mejor máquina de Turing que reproduzca la cadena dada.

Para usar el programa se debe abrir el archivo MT.java en NetBeans u otra plataforma para compilar java y dar play a éste. En la consola aparecerá el siguiente mensaje:

"Enter the text you want to generate with a Turing Machine"

El usuario debe hacer click izquierdo al final del mensaje, escribir la cadena deseada y dar Enter. La aproximación de la Máquina de Turing deseada se iniciará enseguida.