

Tecnologies multimèdia - Avaluacio cont. 4

1. Introducció

El objetivo de este trabajo es la implementación del algoritmo LZ77. El programa acepta una cadena de entrada representando un número binario. También es necesario introducir la longitud de la ventana de desplazamiento y la de entrada. Por defecto el programa comprime la cadena de entrada. Es posible usar el modo de descompresión si se especifica en las opciones.

Se obtiene un string representando una cadena binaria correspondiente a la compresión/descompresión de la cadena binaria de entrada suministrada al programa.

2. Resumen de la implementación

- Implementada la función codificadora especificando los parámetros
 - -i La cadena binaria de entrada
 - -s El tamaño del sliding window
 - -w El tamaño del input window
- Implementada la función decodificadora, es preciso especificar el parámetro **--mode d**.
- Implementada una funcionalidad para realizar tests, es preciso especificar el parámetro **--test** o **-t X**.

Este modo de funcionamiento ejecutará tests con una cadena aleatoria de longitud X y tamaños de sliding window que oscilarán entre 16 y la potencia de 2 más próxima a X , y tamaño de input window desde 16 hasta el tamaño del sliding window en dicha iteración.

En el modo de test es igualmente necesario especificar los parámetros requeridos por el programa (Sliding window, input window) aún cuando no son usados para los tests.

3. Pruebas y resultados

La Figura 1 muestra una de las series de pruebas realizadas, en este caso con una cadena de longitud 900000.

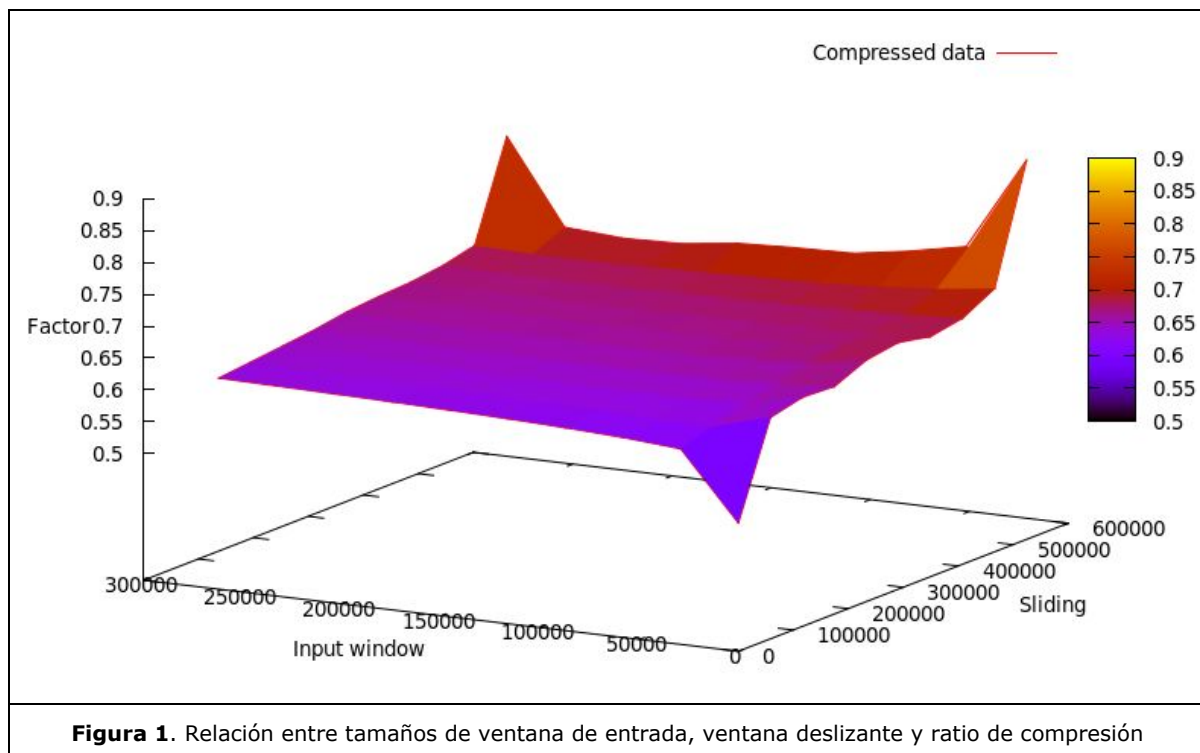


Figura 1. Relación entre tamaños de ventana de entrada, ventana deslizante y ratio de compresión

Se observa que el ratio de compresión es menor de 1 y permanece estable alrededor de 0.6 excepto en las zonas límite donde las ventanas son muy grandes o muy pequeñas. Esto significa que el algoritmo genera una cadena *comprimida* de mayor longitud que la cadena original.

Observamos que el mayor ratio de compresión se obtiene cuando la sliding window tiene el tamaño máximo posible, sin superar el tamaño de la propia cadena de entrada. Esto conlleva que está parte permanezca tal cual en la cadena de salida, sin ser codificada, con lo cual hay menos valores a comprimir y ello hace que el ratio de compresión sea mayor.

En las zonas en que la ventana es muy grande, no pueden extraerse conclusiones válidas sin corregir los datos obtenidos, ya que se está aplicando el algoritmo de desplazamiento de ventana, sobre una porción de la cadena de

entrada no representativa de la totalidad de esta. Pensemos en una ventana de desplazamiento de igual longitud a la cadena de entrada. El algoritmo solo guardará la cadena y finalizará su ejecución.

4. Conclusiones

Aplicando el algoritmo tal y como se nos plantea en el enunciado de la práctica no conseguimos comprimir los datos. Sospechamos que esto se debe a que para codificar cada una de las coincidencias estamos usando una longitud fija.

Probablemente los resultados mejorarían si asociamos este algoritmo a un codificador *Huffman* o algún otro algoritmo de codificación con prefijo.