

TP Organización de datos – Compresor de datos

Grupo 1

Integrantes:

Alic, Mirko
Deciancio, Nicolas
Merino, Gonzalo
Monsech Prada, Santiago

Enunciado:

El objetivo de este TP es realizar un compresor de archivos de propósito general. El método de compresión a usar debe ser creado por el grupo desde cero o bien ser una mejora o modificación a algún método de compresión existente. No se aceptan compresores que sean una implementación “Vainilla” de algún método de compresión conocido.

Modelo:

A partir de una salida de Block Sorting + MTF (Move-to-front), se va a comprimir utilizando un modelo basado en el estructurado.

Suponemos inicialmente que en la salida de un BS+MTF se tiene globalmente una tendencia que indica que el estructurado es el modelo óptimo a utilizar. Sin embargo, localmente dentro de esa salida, esto no se cumple en todos los casos. Puede haber casos, locales, donde el estructurado no sea el modelo óptimo para utilizar.

Lo que nosotros ofrecemos como mejora es que, mientras procesamos el MTF, analizamos una cantidad fija de caracteres y sus frecuencias, y a partir de esto, elegimos un modelo que corresponda mejor, si es posible, que el estructurado, a ese caso en particular.

Algunas de las limitaciones que esto presenta son:

Debemos limitar la cantidad de modelos almacenados dentro del compresor y descompresor, ya que las combinaciones totales para 256 caracteres en 8 niveles, no entrarían en memoria, lo cual lo volvería altamente ineficiente.

En el caso de que el archivo no utilizase ninguno de los modelos alternativos que se adaptarían mejor que el estructurado, por defecto, utilizaría el modelo estructurado en todos los casos y esto nos daría un archivo levemente menos comprimido debido al “overhead” que conlleva guardar la elección de modelo que usamos en cada uno de los casos.

Las ventajas son:

En el caso que tengamos modelos que se adapten mejor a la tendencia presentada dentro de los distintos bloques del archivo (tomando en cuenta el “overhead” antes dicho), esto nos daría un archivo final mas comprimido, que si usáramos el modelo estructurado a secas.

Implementación:

Iniciamos separando en bloques, aplicando sobre estos Block Sorting + MTF. Mientras procesamos el MTF, llevamos un conteo de las frecuencias de los caracteres leídos; cada una cantidad prefijada de caracteres (que sea tal que rompa la tendencia global que favorece al modelo estructurado y ofrezca, en cuanto a lo que comprime, un “overhead” razonable), elegimos el modelo que mejor se adapte, indicando al inicio de esta cadena, con un símbolo, que modelo elegimos. Luego, aplicamos sobre esta cadena el modelo elegido y lo comprimimos con codificación aritmética.

En cuanto al descompresor, funciona de manera análoga, ya que la codificación es única y conoce los modelos utilizados.

Resta definir los parámetros que utilizaremos, particularmente la cantidad de modelos ofrecidos y el tamaño de los bloques que comprimiremos con cada modelo. Esto dependerá de futuras pruebas.