**MÉTODO DE COMPRESSÃO LEMPEL-ZIV**

Luiz Augusto Lima Alves¹

Jones Fernando Giacon2

**1 INTRODUÇÃO**

Primeiramente, podemos definir compressão de dados como um processo de redução do número de bits usados para representar os dados. Existem diversos outros tipos de compressão de dados, neste trabalho veremos um pouco sobre o Lempel-Ziv. Lempel-Ziv é um algoritmo de compressão de dados proposto por J. Ziv e A. Lempel. Eles criaram dois algoritmos chamados também de LZ77 e LZ78, no ano 1977 e 1978, respectivamente. Inicialmente eram conhecidos como LZ1 e LZ2. Esse algoritmo é baseado na utilização de partes de string que já foram lidas de um arquivo baseado em dicionário onde é codificado os símbolos, comprimentos de variáveis, como códigos individuais. O objetivo é identificar cadeias consecutivas de fatores cada vez maiores.

**2 FUNDAMENTOS**

O método Lempel-Ziv utiliza duas formas de trabalho: dicionário, onde ficaram salvas os dados codificados e a área de pesquisa, onde fica a string que será comprimida ou descomprimida. O algoritmo trabalha com a construção do dicionário, compressão dos dados e descompressão dos dados do arquivo.

**2.1 CONSTRUÇÃO DO DICIONÁRIO**

O dicionário inicia com os 256 valores da tabela ASCII. O ficheiro a comprimir é recortado em cadeias de bytes (assim, para imagens monocromáticas - codificadas em 1 bit - esta compressão é pouco eficaz), cada uma destas cadeias é comparada com o dicionário e acrescentada se por acaso não estiver presente.

**2.2 COMPRESSÃO**

Na compressão, o algoritmo percorre a área de pesquisa, e identifica a maior sequência de símbolos armazenada no dicionário e substitui por um código definido, e se por acaso uma cadeia é mais pequena que a maior palavra do dicionário, então é transmitida. A compressão está na substituição das ocorrências destes fatores por um conjunto de índices que mostram sua posição e tamanho na janela deslizante. Esta janela é uma porção contínua e, à medida que os símbolos vão sendo reconhecidos, eles vão sendo acrescentados no seu final, ao mesmo tempo que os do início são excluídos. A janela e o fator a ser reconhecido têm seus tamanhos limitados. Seja N o tamanho máximo da janela, e F o comprimento máximo do fator. A janela será formada por duas partes: uma chamada de janela reconhecida, com comprimento N-F que dá os últimos símbolos do texto fonte que já foram reconhecidos e a outra de janela não-codificada, que tem outros F símbolos que ainda não foram comprimidos. O código resultante será composto por triplas (d, l ,s), onde d é o deslocamento do fator para a borda da janela reconhecida, l é o comprimento do fator, e s é o próximo símbolo do texto original que não pôde ser reconhecido.

**2.3 DESCOMPRESSÃO**

A descompressão é o oposto, os códigos da área de pesquisa devem ser substituídos pelas sequências de símbolos do dicionário, correspondentes. Decodifica-se as fichas mantendo a janela atualizada de maneira análoga ao processo de compressão. À medida que cada ficha é descomprimida, copia-se os símbolos codificados nas fichas para as janelas. Sempre que se encontra uma ficha-frase é feito a consulta a saída apropriada na janela e procura-se a frase com aquele tamanho específico. Quando se encontra uma ficha-símbolo, gera-se um único símbolo armazenado na própria ficha.

**3 CONCLUSÃO**

Com este trabalho foi possível propor uma técnica eficiente de compressão e descompressão de dados utilizando o método Lempel-Ziv. É notório a eficiência e agilidade do método no processo de reduzir tamanho de arquivos (comprimir), bem como voltar ao seu tamanho original (descomprimir).

**4 REFERÊNCIAS**

**Modelos prescritivos para o desenvolvimento de software.** 2012. Disponível em: http://www.galeote.com.br/blog/2012/06/modelos-prescritivos-para-o-desenvolvimento-de-software/. Acesso em: 01/11/2015

PRESSMAN, Roger S. **Engenharia de software**. 7ª ed. Porto Alegre: Bookman, 2009

PRESSMAN, Roger S. **Engenharia de Software.** Mc Graw Hill, 6ª ed. Porto Alegre: 2010.