Enhancing Lempel-Ziv codes using an On-line Variable Length Binary Encoding

Tinku Acharya  
Joseph F. JáJá  

Institute for Systems Research and  
Institute for Advanced Computer Studies  
University of Maryland, College Park, MD 20742  
Email : {acharya, joseph}@umiacs.umd.edu

Abstract

LZW Algorithm is the most popular dictionary-based adaptive text compression scheme [1]. In LZW algorithm, a changing dictionary contains common strings that have been encountered so far in the text. Motivation of this research is to explore an on-line variable-length binary encoding. We apply this encoding to LZW codes for remedy of the problem that we discussed in our earlier paper in DCC'95 [2]. We call it LZWAJ algorithm. We developed a novel methodology for an on-line variable-length binary encoding of a dynamically growing set of integers by mapping the integers into the leaf nodes of a special binary tree called the “phase in binary tree”. This tree has many interesting and useful properties [3]. The length of a path from the root node to any leaf node is upper bound by \( \log_2 n \), where \( n \) is number of leaf nodes in the tree corresponding to \( n \) integers in the set. As a result, length of each of the \( n \) phase in binary codes is less than or equal to \( \log_2 n \). The interesting property of this encoding is that the code of an integer \( i \leq n \) can be generated from the unsigned binary representation of \( i \) and \( n \) only without physically constructing the “phase in binary tree” data structure and very easy to implement. But we have used the “phase in binary tree” in this paper to understand the underlying logic of the encoding. To show the effectiveness of this encoding, we have used it to encode the output of LZW, although the same methodology can be applied in any Lempel-Ziv code. The software has been implemented and tested with different kind of texts. The compression performance of the proposed scheme is much better compared to other known methods.

Acknowledgement: We are indebted to Professor James A. Storer for his valuable suggestions towards the benchmarking of the compression performance. We are also thankful to Professor Amar Mukherjee for his insightful comments in this research.