LZAC Lossless Data Compression

Allan Chu
Saratoga High School
Saratoga, CA 95070

This paper presents LZAC, a new universal lossless data compression algorithm derived from the popular and widely used LZ77 family. The objective of LZAC is to improve the compression ratios of the LZ77 family while still retaining the family’s key characteristics: simple, universal, fast in decoding, and economical in memory consumption.

LZAC presents two new ideas: composite fixed-variable-length coding and offset-difference coding. Composite fixed-variable-length coding combines fixed-length coding and variable-length coding into a single coding scheme. A pre-calculated threshold determines when to use one coding scheme over the other. Offset-difference coding improves the coding efficiency by coding the difference between the offset and the match length instead of the offset when variable-length coding in a non-lookahead case is used. When variable-length coding is used in a lookahead case, offset-difference coding improves the coding efficiency by coding the difference between the match length and the offset instead of the match length.

The literal case codeword of LZAC, <0, literal>, remains the same as that of LZSS. However, the copy case codeword is different. LZAC presents new multiple copy case codewords. When fixed-length coding is used, the codeword is <1, length, offset>. When variable-length coding is used in a non-lookahead case, the codeword is <1, 0, length, offset-difference>. When variable-length coding is used in a lookahead case, the codeword is <1, 0, 0, \mid offset-difference \mid, offset>.

LZAC also creates two new coding tables, \( \lambda_1 \) and \( \lambda_2 \), for variable-length coding. While the \( \lambda_1 \) coding table is used to code the length, offset, and \( \mid offset-difference \mid \), the \( \lambda_2 \) coding table is used to code the offset-difference.

The two new concepts, composite fixed-variable-length coding and offset-difference coding, enable LZAC to further improve the LZ family’s compression ratios without using the complex and slow adaptive arithmetic coding or Huffman coding. The Calgary corpus was used to evaluate LZAC’s performance. The results show that LZAC achieves a non-weighted average compression ratio of 2.94 bits/char in an 8-kbyte window which surpasses the performance of any other compression algorithm in the LZ77 and LZ78 families.