

## **Exact Data Compression Using Hierarchical Dictionaries**

**Daniel K. Chang**

**The Graduate School and University Center  
The City University of New York**

### **Abstract**

This paper describes a new data compression/expansion algorithm which overcomes some of the shortcomings of the best algorithms known. The algorithm uses multiple (two will be used as an example) dictionaries for storing previously encountered strings of the data file to be compressed. The first dictionary is a short one, it has 128 entries and needs 7 bits to store its address. The second dictionary is a long one, it has 32,768 entries and requires 15 bits to store its address. Each of these two dictionaries is initialized with frequently occurring strings. An input data file is compared with the dictionaries so that the longest substring which can be found in a dictionary will give its position in the dictionary as an output code. Compression and expansion use the same procedure to form and update dictionaries. A compressed result consists of a series of pointers to the dictionaries with a byte as a pointer to the short dictionary and two bytes as a pointer to the long dictionary. During compression/expansion, the more frequently occurring strings will be dynamically swapped into the short dictionary. The two dictionaries are used as scratch pads during the time of program execution and need not be stored/transmitted otherwise.