THE LEMPEL-ZIV-WELCH ALGORITHM

Student IONUȚ FLOREA
Student DRAGOȘ SARCHIZIAN
Student ELENA STOENESTEANU
Student VIRGINIA STOICA
Faculty of Economic Cybernetics, Statistics and Informatics,
Academy of Economic Studies

1.Introduction

LZW compression replaces character rows by simple codes without making the next words analysis. Instead, it adds each new row in a row table. The compression appears when only a code is printed instead of a character row. It starts with a simple table, which associates one code to each possible character (256 inputs of 8 bytes each of them, for example). During the process carrying on, the algorithm is building a larger and more efficient codes table, adding new codes for each new sequence of characters. All that is necessary is to set the table maximum length so that to know the maximum length of the code that can be allocated. The code that LZW algorithm is displaying is of an arbitrary length, but it should consists of more bytes than a single character. The first 256 codes (when 8 bytes characters are used) are implicitly associated to the character standard set. The remaining codes are associated to some rows as long as the algorithm advances. The optimum version is running with 12 bytes codes.

2. Presentation of the Compression Algorithm

LZW compression algorithm consists of the following steps:

1. The dictionary initializing – all the tree nodes are marked as being initialized. All the ASCII 256 characters represent the first 256 symbols of the dictionary, too. The current sequence to be processed is null;
2. The variable NextCode is initialized with the first available value (for the first cycle 257);
3. Character = the next symbol from the input file;
4. The up-dated combinations with the new character are searched in the dictionary; if they do not exist, the formed combination is added in the dictionary with a new code;
5. If exist, the code corresponding to that combination becomes the new current code;
6. Have it finally reached the end of a file? If not, the step 2 is repeated; if yes, the combination is sent, which is the last row of the file;
7. EndOfStream is sent, which informs the decoder that it reached the end of file.

3. Presentation of the Decompression Algorithm

At decompression, the flow of data sent to compression, which is used to rebuild the initial data flow. The only problem to remake the tree appears when the decoded characters are maintained in reverse order, which requires their putting in pile, extracting in reverse order and their writing in the output file.

The algorithm steps are as follows:

1. Initializing-locating of the NextCode variable (used for the managing of the codes added in the dictionary;
2. The code value is read, the row is searched in the dictionary and it is found and is sent;

3. A new row is formed from the OldCode and also the first read character from the current row;
4. If the input flow has not been ended, you should come back to step 2.

Only one exception in the compression algorithm can create decompression problems. Any time when a new row is added in the dictionary at compression, this means that the whole row has been met in the dictionary up to that moment. If by any reason, the compressor has used this row as an equivalence row, problems will be met at decompression. This would mean the decompression of a row, which is not in the dictionary yet. If an existing row in the dictionary consists of a pair Row-Row and an input row, the sequence Caracter+Row+Caracter+ Row+ Character will be met, the compression algorithm will send a code before the decompressor should define it.

These exception treating was made by adding once again the first character of the row at the end.

4. Experimental Results

A trial row is used in the next table to demonstrate the algorithm. Following the steps of the algorithm for this row of characters, it can be noticed that at the first cycling of the loop a check is made to notice if the row "/W" is in the table. If not, the code for "/" is printed and the row "/W" is added to the table. Due to the fact that 256 characters have already been defined with the codes 0-255, the code 256 is associated to the first row defined. After the reading of the third letter "E", the code of the second row "WE" is added to the table, and the code for letter "W" is printed. This continues until the characters '/' and 'W' are read in the second word, matching to the code 256. In this case, code 256 is printed, and a row consisting off three characters is added in the table. The process continues until the row has been exhausted and all codes were printed.

Table 1: Experimental results

Entered row = /WED/WE/WEE/WEB/WET			-
Read code	**Resulted code**	**New code value**	**Adding in the dictionary**
/W	/	256	/W
E	W	257	WE
D	E	258	ED
/	D	259	D/
WE	256	260	/WE
/	E	261	E/
WEE	260	262	/WEE
/W	261	263	E/W
EB	257	264	WEB
/	B	265	B/
WET	260	266	/WET
EOF	T	-	-

The resulted example for the row is presented together with the row table. The row table fills quickly as soon as a new row is added a resulting code to the table. In this characters repeating sequence, five code replacements were made, together with 7 characters. In the examples consisting of real rows, the compression does not start until an adequate table has not been built, usually after at least 100 octets reading.

4. Optimizing

Using of variable codes lengths up to 15 bytes allows a lot of improvements.

The compression rate will increase if it is possible to memorize more rows, so a greater size of the dictionary. For a 15 bytes code maximum length, a dictionary of 32768 rows dictionary can be used. The code length size can lead to increasing the calculation time.

This advantage becomes a disadvantage at the smaller sizes files. To solve this problem, the next program starts by using codes having 9 bytes length, and then passing to 10 bytes at the moment when 256 free inputs were entered the dictionary. Further, it will be used 10, 11, up to 15 bytes.

For the decompression program, a passing code from a level to another will be used, named CodSinc. This code asks the decompression program to modify immediately the code length.

For the large files, the dictionary may fill up at a moment. At this moment, it is canceled and it is rebuilt from 0. This moment is indicated to the decompressor by the filling-up code.

5. Results and Conclusions

The results from the table 2 have been got further to a test on a sample of 12 test files of different types and sizes.

Table 2: Results of applying LZW algorithm

File name	File length (KB)	Compressed	Compression rate(%)	Compressed	Compression rate(%)
Test01.doc	37	17.8	52	12.8	66
Test02.jpg	26	34.1	-31	32.7	-25
Test03.exe	46	65	-41	61.7	-34
Test04.dat	3	1.17	61	0.97	68
Test05.txt	3	1.74	42	1.48	51
Test06.htm	11	5.49	50	5.15	54
Test07.ani	12	2.60	79	2.26	82
Test08.obj	3	2.62	13	2.28	24
Test09.gif	3	3.63	-21	3.29	-9
Test 10.z80	45	54.2	-20	39.2	13
Test11.pod	3	0.99	64	0.82	73
Test12.bmp	1407	20	98	24	98

- The best application for LZ algorithms is data memorizing, which are used quite often, without being modified each time (statistical annual books, phone books…). Also, the files with many recurrent data use this algorithm to develop complex applications. Test12.bmp file has experimentally been created by 50 times multiplication of a figure, but generally, this type of file is very well compressed by Lempel-Ziv algorithm class.
- Taking into account the large using of LZ algorithm in data sending, the method has been licensed. The holder of the license on this algorithm, Unisys Company, allows the free use of this method, excepting the commercial software makers.

References

1. [IVAN98] Ion Ivan, Daniel Verniş " Data Compression", Cison Publishing House, Bucharest, 1998
2. [IVAN96] Ion Ivan, Daniel Vernis "Data Compression"
   "PC Report" No.48, September 1996, page 71-73
3. Data Compression Reference
   Center http://www.rasip.fer.hr/research/compress/algorithms/index.html
4. Arturo San Emeterio Campos "LZW decoding"
   http://www.arturocampos.com/ac-lzw-gif.html
5. http://www.data-compression.com/ - the official site of data compression.