International Workshop On
Combinatorial Algorithms
IWOCA

http://www.iwoca.org—Problems section.

Can Lempel-Ziv and Burrows-Wheeler compression be asymptotically compared?

Nicola Prezza
University of Udine, Department of Mathematics, Physics, and Computer Science
prezza.nicola@spes.uniud.it

presented at IWOCA 2016
18 Aug. 2016

1 Lempel-Ziv factorization

The Lempel-Ziv factorization [5] (LZ77) of a \$-terminated string $T \in \Sigma^n$ (\$ symbol not appearing elsewhere in T) is obtained by factoring T in z phrases, each phrase being the shortest factor that does not appear before in the text. For example,

$$LZ77(babbababbabba\$) = b|a|bb|aba|bbabb|a\$|$$

In the above example, the number z of LZ77 phrases is $z = 6$.

2 Burrows-Wheeler Transform

The Burrows-Wheeler transform [1] (BWT) of a \$-terminated string $T \in \Sigma^n$ (\$ character not appearing elsewhere in T and lexicographically smaller than all other alphabet characters) is a permutation of T obtained by sorting all circular permutations of T in a matrix of size $|T| \times |T|$ (having T's circular permutations as rows) and by taking the last column of this matrix. Figure 1 depicts this matrix for the string *babbababbabba*\$; taking the last column, we obtain:

$$BWT(babbababbabba\$) = abbbbbbbb\$aaaa$$

$BWT(T)$ is a reversible permutation and can be efficiently compressed with *run-length encoding*, i.e. by replacing it with the shortest list of pairs $\langle c_i, \ell_i \rangle_{i=1,\ldots,r}$, $c_i \in \Sigma$, $\ell_i \in \mathbb{N}$ such that $BWT(T) = c_1^{\ell_1} c_2^{\ell_2} \ldots c_r^{\ell_r}$. In the above example, this list is $\langle a, 1\rangle$, $\langle b, 8\rangle$, $\langle \$, 1\rangle$, $\langle a, 4\rangle$ (with $r = 4$).

```
$babbababbabba
a$babbababbabb
ababbabba$babb
abba$babbababb
abbababbabba$b
abbabba$babbab
ba$babbababbab
bababbabba$bab
babba$babbabab
babbababbabba$
babbabba$babba
bba$babbababba
bbababbabba$ba
bbabba$babbaba
```

Figure 1: Burrows-Wheeler matrix for the string *babbababbabba*$

3 The problem

Lempel-Ziv- and (run-length encoded) BWT- based compressors output compressed representations of T taking, respectively, $\mathcal{O}(z)$ and $\mathcal{O}(r)$ words of space. Both z and r are important measures of repetitiveness of T—being closely related to its number of self-repetitions—and can be (up to) exponentially smaller than $|T|$. A very interesting open problem—first addressed in [4]—is how the two measures relate to each other.

Let $\Sigma = \{s_1, \ldots, s_\sigma\}$ be the alphabet. Both z and r are at least σ and can be $\Theta(\sigma)$, e.g. in the text $(s_1 s_2 \ldots s_\sigma)^e$, $e > 0$. However, the rate r/z can be $\Theta(\log_\sigma n)$: this happens, for example, in de Bruijn sequences[1] of order $k > 1$.

Conversely, also the rate z/r can be $\Theta(\log n)$. This is the case, e.g., of Fibonacci words, which are defined recursively as follows: $f_1 = a$, $f_2 = b$, $f_n = f_{n-1} f_{n-2}$. The string *babbababbabba*$ in the above examples is f_7 (terminated by $). Fibonacci words are a particular case of *standard words*; such words produce a total clustering of the alphabet letters in the BWT [3] (i.e. two runs), and represent therefore one of the cases where the BWT can be compressed to just $\mathcal{O}(\log n)$ bits. On the other hand, the LZ77 factorization of f_n corresponds to the factorization of f_n into *singular words* $\hat{f}_i$, where each $\hat{f}_i$ is obtained by complementing the first letter in the left rotation of the Fibonacci word f_i (see [2] for more details). Since $|f_i|$ is exponential in i, it follows that the Lempel-Ziv factorization of f_n has $\Theta(\log |f_n|)$ factors.

To the best of our knowledge, no examples where the above rates asymptotically exceed $\Theta(\log n)$ are known. An interesting problem is therefore that of establishing whether r and z can always be compared (up to a small $\mathcal{O}(\log n)$ factor):

[1] To see this, consider the BWT row-partition induced by length-$(k-1)$ strings in the first $k-1$ columns of the matrix. Each $x \in \Sigma^{k-1}$ appears exactly σ times in the de Bruijn sequence and all such occurrences are preceded by different characters. It follows that each of the above BWT partitions contains at least $\sigma - 1$ runs, so the BWT has at least $(\sigma - 1)\sigma^{k-1} \in \Theta(\sigma^k) = \Theta(n)$ runs. The number of LZ77 phrases of any text is, on the other hand, always $\mathcal{O}(n/\log_\sigma n)$.

Conjecture 1 *The numbers r of equal-letter runs in the Burrows-Wheeler transform and z of Lempel-Ziv (LZ77) phrases of any length-n text always satisfy*

$$r/z + z/r \in \mathcal{O}(\log n)$$

References

[1] Michael Burrows and David J Wheeler. A block-sorting lossless data compression algorithm. 1994.

[2] Gabriele Fici. Factorizations of the Fibonacci infinite word. *Journal of Integer Sequences*, 18(2):3, 2015.

[3] Sabrina Mantaci, Antonio Restivo, and Marinella Sciortino. Burrows–wheeler transform and sturmian words. *Information Processing Letters*, 86(5):241–246, 2003.

[4] Jouni Sirén et al. *Compressed full-text indexes for highly repetitive collections.* PhD thesis, 2012.

[5] Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, 23(3):337–343, 1977.