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 Pisa, Italy
nicola.prezza@gmail.com

presented at IWOCA 2016
18-8-2016

1 Lempel-Ziv factorization

The Lempel-Ziv factorization [10] (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 [9]—is how the two measures relate to each other.

Let $\Sigma = \{s_1, \dots, s_\sigma\}$ be the alphabet. Both z and r are at least σ and can be $\Theta(\sigma)$, e.g. in the text $(s_1 s_2 \dots s_\sigma)^e$, $e > 0$. 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 [7] (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. It seems therefore natural to conjecture that the ratios r/z and z/r are always $\mathcal{O}(\log n)$.

[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 $\mathcal{O}(n/\log_\sigma n)$.

3.1 Recent Developments

Recently, one direction of the problem has been solved. In [5], the authors showed that $z \in \mathcal{O}(r \log^2(n/r))$ using the recent notion of *string attractor*. This bound has been improved to the optimal $z \in \mathcal{O}(r \log(n/r))$ in [3] using grammars based on locally-consistent parsing. This upper-bound is tight since, as observed in the previous section, Fibonacci words satisfy $z/r \in \Theta(\log n)$.

As far as the other direction is concerned, Pape-Lange showed in [8] that $r \in O(z^2 \log n)$. Kempa and Kociumaka [4] improved this bound to $r \in O\left(z \log z \max(1, \log \frac{n}{z \log z})\right)$. In the same paper, they actually prove $r \in O\left(\delta \log \delta \max(1, \log \frac{n}{\delta \log \delta})\right)$, where $\delta \leq z$ is a stronger measure of repetitiveness recently studied in [6], and prove the bound to be tight for all values of n and δ. While this essentially solves the present conjecture as a function of δ, the tightness of the bound as a function of z is still open:

Question 1 *Is the bound* $r \in O\left(z \log z \max(1, \log \frac{n}{z \log z})\right)$ *tight?*

Pape-Lange speculates in [8] that r may be upper-bounded by a polynomial in z.

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] Travis Gagie, Gonzalo Navarro, and Nicola Prezza. On the approximation ratio of lempel-ziv parsing. In *Latin American Symposium on Theoretical Informatics*, pages 490–503. Springer, 2018.

[4] Dominik Kempa and Tomasz Kociumaka. Resolution of the Burrows-Theeler transform conjecture. *Proceedings of IEEE 61st Annual Foundations of Computer Science*, 2020.

[5] Dominik Kempa and Nicola Prezza. At the roots of dictionary compression: String attractors. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 827–840, New York, NY, USA, 2018. ACM.

[6] T. Kociumaka, G. Navarro, and N. Prezza. Towards a definitive measure of repetitiveness. In *Proc. 14th Latin American Symposium on Theoretical Informatics (LATIN)*, 2020. To appear.

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

[8] Julian Pape-Lange. On extensions of maximal repeats in compressed strings. *Proceedings of the 31st Annual Symposium on Combinatorial Pattern Matching*, 2020.

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

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