Indexing the Bijective BWT

Hideo Bannai[1], Juha Kärkkäinen[2], Dominik Köppl[3], and Marcin Piątkowski[4]

[1] Department of Informatics, Kyushu University, Fukuoka, Japan
[2] Helsinki Institute of Information Technology (HIIT) and
Department of Computer Science, University of Helsinki, Finland
[3] Department of Computer Science, TU Dortmund, Dortmund, Germany
[4] Faculty of Mathematics and Computer Science,
Nicolaus Copernicus University, Toruń, Poland

bannai@inf.kyushu-u.ac.jp juha.karkkainen@cs.helsinki.fi
dominik.koeppl@tu-dortmund.de marcin.piatkowski@mat.umk.pl

Abstract. We propose a self-index that works like the FM-index [3], but is built on the bijective BWT instead of the BWT. Like the FM-index, the index supports efficient backward searching.

The Burrows-Wheeler transform (BWT) [1] is a reversible transformation permuting all symbols of a given string s. The output is formed by the characters preceding each suffix in the lexicographical order of all suffixes of s. Such an operation tends to group identical characters together, which has many applications in data compression.

Notice that all conjugates (cyclic rotations) of a given string share the same BWT. Moreover, some strings cannot be considered as valid BWTs (e.g. `bccaab` cannot be reversed). However, one can consider a bijective version of BWT [4, 5] based on the Lyndon factorization [2] of the input string. In this case the output consists of the last symbols of the lexicographically sorted cyclic rotations of all Lyndon factors of the input. Since each string has a unique factorization into lexicographically nonincreasing Lyndon words such a transformation induces a bijection between strings of a given length n and multisets of Lyndon words of total length n.

A text index is a data structure built over an input text t allowing fast search for all occurrences of a given pattern p without the need of traversing the whole text. Many text indices contain sufficient information to retrieve any substring of t such that it can replace the text itself to reduce the space consumption. One efficient kind of these indices [6] is built on the traditional BWT. In this light, one may ask whether it is possible to build similar index data structures by exchanging the traditional BWT with the bijective variant. We answer this question affirmatively, presenting the first index built on the bijective BWT.

References

1. M. Burrows and D. J. Wheeler. A block sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation, Palo Alto, California, 1994.
2. K. T. Chen, R. H. Fox, and R. C. Lyndon. Free differential calculus, IV. The quotient groups of the lower central series. *Annals of Mathematics*, pages 81–95, 1958.
3. P. Ferragina and G. Manzini. Indexing compressed text. *J. ACM*, 52(4):552–581, 2005.
4. J. Y. Gil and D. A. Scott. A bijective string sorting transform. *arXiv*, abs/1201.3077, 2012.
5. M. Kufleitner. On bijective variants of the Burrows-Wheeler transform. In *Proc. PSC*, pages 65–79, 2009.
6. G. Navarro and V. Mäkinen. Compressed full-text indexes. *ACM Comput. Surv.*, 39(1):article 2, 2007.