International Workshop On Combinatorial Algorithms
IWOCA

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

Open problems on prefix normal words

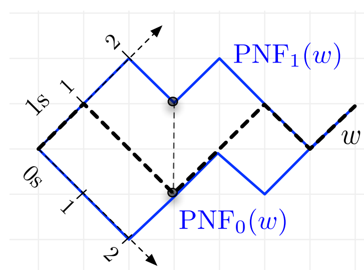
Zsuzsanna Lipták
University of Verona, Italy
zsuzsanna.liptak@univr.it

presented at IWOCA 2014
15 Oct. 2014

A binary word is called *prefix normal w.r.t. 1* if no substring has more 1's than the prefix of the same length. For example, 1001101 is not prefix normal because the substring 11 has two 1's while the prefix 10 has only one. Prefix normal w.r.t. 0 is defined analogously. These words were introduced in [1]. They occur within the context of Binary Jumbled Pattern Matching (BJPM, see below) but are also of independent interest.

For a binary word w, let $F_1(w,k)$ denote the maximum number of 1's in any substring of length k. $F_0(w,k)$ is defined analogously. Below (left) we give the functions $F_1(w,\cdot)$ and $F_0(w,\cdot)$ for the word $w = 1001101$.

k	0	1	2	3	4	5	6	7
$F_1(w,k)$	0	1	2	2	3	3	3	4
$F_0(w,k)$	0	1	2	2	2	3	3	3



In [1] it was shown that for every word w, there is exactly one prefix normal word (w.r.t. 1) w' s.t. $F_1(w,\cdot) = F_1(w',\cdot)$, which we call w's *prefix normal form w.r.t.* 1, $\text{PNF}_1(w)$. For example, $\text{PNF}_1(1001101) = 1101001$. In actual fact, $\text{PNF}_1(w)$ is the function of differences of $F_1(w)$, viewed as a binary word. More precisely, note that $F_1(w,k+1)$ is either equal to $F_1(w,k)$, or is $F_1(w,k)+1$. If we write, for $k = 1,\ldots,n$, a 1 when $F_1(w,k+1) = F_1(w,k)+1$, and a 0 otherwise, then we get the desired prefix normal word $w' = \text{PNF}_1(w)$. Again, the analogous construction yields w's prefix normal form w.r.t. 0, $\text{PNF}_0(w)$, in our example $\text{PNF}_0(w) = 0011011$.

In the figure above (right), we display binary words as starting from the origin, and putting a unit segment towards NE for a 1 and towards SE for a 0. The bold dashed line shows $w = 1001010$ and the blue lines its two prefix normal forms.

Open Problems:

1. *Given a binary word w, compute its prefix normal form (w.r.t. 1), fast (see below).*
2. *Given a binary word w, test whether it is prefix normal (w.r.t. 1), fast (see below).*

Both can be solved simply in $O(n^2)$ time, where n is the length of w, by first computing the F_1-function of w, and then $\mathrm{PNF}_1(w)$, as demonstrated above. Note that this also solves the testing problem, since a word is prefix normal (w.r.t. 1 resp. 0) iff it is equal to its prefix normal form (w.r.t. 1 resp. 0). Or one can use one of several recent solutions for BJPM (see below), which compute the F_1-function using reduction to min-plus convolution in some form, in $O(n^2/\text{polylog } n)$ time [4, 5], or recently even somewhat faster, in $n^2/2^{\Omega(\log n/\log\log n)^{1/2}}$ time [6]. We are interested in either

- faster solutions than these (for *1.*, this would also improve BJPM solutions); or
- $o(n^2/\log n)$ time solutions which do not first compute the F_1-function. The hope would be to find a way of computing the prefix normal form directly on a string level.

To find out more about prefix normal words, see [1, 2, 3], which also list more open problems. The latter two papers contain a combinatorial generation (listing) algorithm for, resp. some combinatorial results on enumeration of, prefix normal words; however, they have not brought us closer to the solution of the above problems (the testing algorithm in [3] has quadratic worst-case time).

Postscriptum: **Binary Jumbled Pattern Matching (BJPM)** is the following problem: Given a binary word w, create an index for w such that queries of the following form can be answered quickly: Does w have a substring with x number of 1's and y number of 0's? All current solutions compute a linear size index for w, which can then be queried in constant time. The connection to prefix normal words is the following:

> Given w and a BJPM query (x, y). Then the answer is YES if and only if the prefix of length $x + y$ of $\mathrm{PNF}_0(w)$ has at least x 1's, and the prefix of length $x + y$ of $\mathrm{PNF}_1(w)$ has at most x 1's.

Indeed, in the figure we see that all substrings of length 3 have either two or one 1's (vertical line).

The BJPM problem has attracted a lot of interest recently, for a current overview see my talk given at McMaster University in October 2014 (`profs.scienze.univr.it/~liptak`).

References

[1] G. Fici and Zs. Lipták: On prefix normal words. *Proc. of the 15th Int'l Conf. on Developments in Language Theory (DLT 2011)*, LNCS 6795:228–238, 2011.

[2] P. Burcsi, G. Fici, Zs. Lipták, F. Ruskey, and J. Sawada: On Combinatorial Generation of Prefix Normal Words, *Proc. of the 25th Ann. Symp. on Combinatorial Pattern Matching (CPM 2014)*, LNCS 8486:60-69 (2014).

[3] P. Burcsi, G. Fici, Zs. Lipták, F. Ruskey, and J. Sawada: Normal, Abby Normal, Prefix Normal, *Proc. of the 7th Int'l Conference on Fun with Algorithms (FUN 2014).* LNCS 8496:74–88 (2014).

[4] P. Burcsi, F. Cicalese, G. Fici, and Zs. Lipták: On Table Arrangements, Scrabble Freaks, and Jumbled Pattern Matching, *Proc. of the 5th Int'l Conference on Fun with Algorithms (FUN 2010).* LNCS 6099:89–101 (2010).

[5] T.M. Moosa and M.S. Rahman: Sub-quadratic time and linear space data structures for permutation matching in binary strings. *J. Discrete Algorithms* 10:5–9 (2012).

[6] D. Hermelin, G. Landau, J. Rabinovits and O. Weimann: Binary Jumbled Pattern Matching via All Pairs Shortest Paths. *Corr abs1401.2065v4* (2014).