A Brief List Of Thesis Topics (Bachelor)

Enno Ohlebusch[*]

March 7, 2018

1 The k-common substring problem

Given m strings, the *k-common substring problem* is to compute a table CS of size m, where $CS[k]$ stores a longest substring common to at least k of the strings. This problem has been solved by Hui [6]: he used a constant-time solution to the Lowest Common Ancestor (LCA) problem in combination with suffix trees. Crochemore et al. [3] showed that the problem can also be solved with a union-find data structure. Your task is to explain the two different solutions and to implement the algorithm of Crochemore et al. (using an existing implementation of a union-find data structure).

2 Position heaps

Ehrenfeucht et al. [4] address the *pattern matching problem*, i.e., the problem of finding the locations of all instances of a string P in a text T, where preprocessing of T is allowed in order to facilitate the queries. Their solution uses a data structure called *position heaps*. Chairungsee and Crochemore [2] use an augmented position heap to compute the Longest Previous non-overlapping Factor (LPnF) table. The LPnF table has applications in string algorithms and data compression. Your task is to explain position heaps and to implement the algorithm of Chairungsee and Crochemore (using an existing implementation of position heaps, see e.g. `http://www.cs.colostate.edu/PositionHeaps/`).

3 Pattern matching with child tables

Pattern matching can also be done with child tables [1]. A child table is a fundamental way to achieve fast look-up in an index data structure (the suffix array of text T). Frith and Shrestha [5] provide a new description of child tables and demonstrate their generality. They also show that child tables can be used without LCP (longest common prefix) tables, reducing the memory requirement. Your task is to explain and implement child tables, and to compare their practical performance with binary search for pattern P on the suffix array of T.

[*] Institute of Theoretical Computer Science, University of Ulm, D-89069 Ulm, enno.ohlebusch@uni-ulm.de

References

[1] M.I. Abouelhoda, S. Kurtz, and E. Ohlebusch. Replacing suffix trees with enhanced suffix arrays. *Journal of Discrete Algorithms*, 2(1):53–86, 2004.

[2] S. Chairungsee and M. Crochemore. Longest previous non-overlapping factors table computation. In *Proc. 11th International Conference on Combinatorial Optimization and Applications*, volume 10628 of *Lecture Notes in Computer Science*, pages 483–491. Springer, 2017.

[3] M. Crochemore, C.S. Iliopoulos, A. Langiu, and F. Mignosi. The longest common substring problem. *Mathematical Structures in Computer Science*, 27(2):277–295, 2017.

[4] A. Ehrenfeucht, R.M. McConnell, N. Osheim, and S.-W. Woo. Position heaps: A simple and dynamic text indexing data structure. *Journal of Discrete Algorithms*, 9(1):100–121, 2011.

[5] M.C. Frith and A.M.S. Shrestha. A simplified description of child tables for sequence similarity search. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2018. To appear, see `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8288582`.

[6] L.C.K. Hui. Color set size problem with applications to string matching. In *Proc. 3rd Annual Symposium on Combinatorial Pattern Matching*, volume 644 of *Lecture Notes in Computer Science*, pages 230–243. Springer, 1992.