Compressed data structures for strings

Paolo Ferragina

Dipartimento di Informatica, Università di Pisa, Italy
ferragina@di.unipi.it
www.di.unipi.it/~ferragin

Abstract

I have seen too many papers that, when dealing with large sets of long strings, argue that one must choose between the *time* efficiency of some (more or less sophisticated) search operations and the *space* succinctness of their solutions. In my lectures I address this *"old" dichotomy* by discussing the most recent advances in compressing and indexing strings. The moral is that: theoretically, it is no longer the case that such a dichotomy does exist; practically, do exist some engineered implementations that ease and stimulate the use of these sophisticated and effective tools in real applications.

On the Lectures

The string matching field has grown to such a complicated stage that various issues come into play when studying it: data structure and algorithmic design, compression techniques, architectural features, database principles, algorithmic engineering and experimentation. My lectures concentrate on the currently well-studied interplay that does exist between two research fields: indexing data structures and compressor design. This connection, at a first glance, might appear paradoxical because these "tools" have antithetical goals. In fact, index design aims at augmenting data with routing information (i.e. data structures) that allow the efficient retrieval of patterns or the extraction of some information. Conversely, compressors aim at removing the repetitiveness present in the data to squeeze them in a reduced space occupancy. Recent results have shed new light on these two fascinating topics by surprisingly showing that compressed indexes and strong compressors do exist, and they can be designed by carefully orchestrating known, and novel, ideas born in both these two research fields. This is actually an active area of research that, apart of interesting solutions to many individual problems, lead to a foundational contribution: several indexing and compression problems *can be reduced to* the design of some surprisingly simple basic tools; improving these tools immediately leads to *guaranteed* time and space improvements for the more sophisticated problems. This algorithmic framework has a twofold advantage: theoretically, it allows researchers to study the simpler problems in order to design efficient solutions and/or derive computational limitations for them; practically, it allows programmers to build efficient compressed indexes starting from engineered implementations of those basic blocks (see e.g. the Pizza&Chili site [5]).

To highlight these interesting algorithmic issues, I deal with strings of various types— binary or from a general alphabet, raw or with some structure (e.g. XML)— and with the design of several kinds of query operations— from the classical substring/prefix/suffix searches to more sophisticated operations which involve string content and structure. The following bibliography provides few seeds to start digging into this fascinating topic.

References

[1] J. Barbay and I. Munro. Succinct Encoding of Permutations and its Applications to Text Indexing. *Encyclopedia of Algorithms*, Editor in Chief Ming-Yang Kao, Springer, 2007 (to appear).

[2] P. Ferragina, R. Giancarlo, G. Manzini, M. Sciortino. Compression boosting in optimal linear time. *Journal of the ACM*, 52(4):688-713, 2005.

[3] P. Ferragina, R. Giancarlo, G. Manzini. The myriad virtues of wavelet trees. *International Colloquium on Automata, Languages and Programming*, LNCS vol. 4051, 561–572, 2006.

[4] P. Ferragina, F. Luccio, G. Manzini, S. Muthukrishnan. Structuring labeled trees for optimal succinctness, and beyond. *IEEE Symposium on Foundations of Computer Science*, 184–196, 2005. Extended version downloadable from `http://roquefort.di.unipi.it/~ferrax/xml_ferra.pdf`

[5] P. Ferragina and G. Navarro. The PIZZA&CHILI site. Two mirrors at `http://pizzachili.di.unipi.it` and `http://pizzachili.dcc.uchile.cl`.

[6] P. Ferragina, R. Venturini. A simple storage scheme for strings achieving entropy bounds. *Theoretical Computer Science*, 372(1): 115-121, 2007.

[7] P. Ferragina, R. Venturini. Compressed permuterm index. *ACM SIGIR Conference*, 2007 (to appear).

[8] A. Gupta, W.K. Hon, R. Shah, J. Vitter. Compressed data structures: dictionaries and data-aware measures. *IEEE Data Compression Conference*, 213-222, 2006.

[9] G. Navarro and V. Mäkinen. Compressed full text indexes. *ACM Computing Surveys*, 39(1), 2007.

[10] N. Raman and R. Raman. Rank and select operations on binary strings. *Encyclopedia of Algorithms*, Editor in Chief Ming-Yang Kao, Springer, 2007 (to appear).